DETECTION AND DEFENSE APPROACHES FOR SENSITIVE DATA LEAKAGE IN CLOUD COMPUTING

CSE 543: Information Assurance and Security

Spring 2020, Group 2

Srinivasula Reddy Kalluri (L) - 1216230808

Rishti Gupta (DL) - 1217211814

Sai Hemanth Gantasala - 1215187051

Krishna Sumanth Gundluru - 1211331898

Vishwarajsinh Sodha - 1217485776

Nachiketa Pathak - 1217117577

Akshay Agarwal - 1216766629

Venkatraman Balasubramanian - 1213345287

Avinash Khatwani - 1216635004

Pranjali Dileep Jagtap -1214157908

Ira A. Fulton Schools of Engineering

Arizona State University

SUMMARY

With the growth of modern technologies and an extensive amount of data that gets generated on a daily basis, cloud has become one of the major resources used by both private organizations and the public users to store and retrieve the data. The paper focuses on data leakage which is a threat that can eventually cause major problems like data theft, security breach, identification theft, unauthorized access etc. Hence, it is very essential to have a proper defence mechanism against the data leakage and also to be able to detect the data leakage as soon as it happens so that proper actions can be taken to retrieve the leaked data and prevent further loss. In this project, we extensively studied the research papers related to data leakage in cloud computing and have logically presented our findings here. We start with the discussion of various types of cloud computing models and services. We identify the threats and vulnerabilities faced by cloud computing, narrowing our discussion about the particular threat of data leakage. We discuss the various ways in which the data on the cloud can be susceptible to the leakage. In the next two sections, we discuss the techniques that can help prevent the data leakage and then a few techniques to detect the data leakage in case it happens. We conclude our report by analysing our findings and comparing various techniques.

Table of Contents

INTRODUCTION

1.1 Background and Motivation

Cloud computing is an innovation methodology which is based on the principle of pay per utilize access which enables gathering of shared resources for stockpiling, servers, specific systems, administration and applications, with no physical access to the cloud. Numerous businesses use this facility to improve their efficiency by transferring data from one location to another, for example, storing data related to keeping money, social insurance and instruction.

The primary objective of information security is not to share the data to unapproved elements. In any case, it's not constantly conceivable to forestall information spillage since we need to access and offer to utilize that data. Typically, information spillage happens while getting to and sharing the data. Business will get affected when touchy data, for example, client subtleties and exchange subtleties is spilled to the individuals who are in the same business. Government data leaks may include sensitive information, for example, international strategies and interior security and military subtleties. These days where whole data is put away in the cloud we have to address security issues, since each application is putting away their information in the cloud and it has all the data identified with client individual data, for example, email address and information of birth and their perusing history. The day was not far when data breach happened in one of the Leading Software Companies, Facebook - considered to be having the most robust Data Security Systems on Cloud. It shocked a lot of people on how risky, data leakage can be. According to Facebook, there were at least 50 million users' personal data which was at risk. The attackers used Facebook Developer APIs to obtain profile specific information. Although payment related information was not stolen, neither were personal messages accessed. But this was enough damage to gear up security systems and security measures. Across the world it has become one of the most hot topics and important fields. This was the primary motivation that enabled us to choose the topic for Detection and Defense Approaches for Sensitive data Leakage in Cloud Computing.

In recent years, work has been dedicated to develop security framework and mechanisms to protect user data. Although security is analyzed from different perspectives, more attention is given for enhancing security and actual data leakage issues are not considered. Our focus is to dig deep into the security issues of data breaches and their significance in data security.

As a lot of data, sensitive or otherwise, is being stored on the cloud, detection and prevention of attacks on the data become very important for data security. We wish to analyze the data leakages in cloud computing motivated by recent data breaches that happened at Facebook, iCloud, Sony, etc.

1.2 Goals and Scope

In this project, we will be investigating and classifying different detection and defensive mechanisms based on their characteristics in all layers of the cloud system of sensitive data leakage in cloud computing. We aim to evaluate the various defense techniques and provide a comparative analysis. We will also discuss any potential solutions to the unresolved drawbacks to data security in cloud computing systems. In the current business scenario, data leakage is a big challenge as critical organizational data should be protected from unauthorized access. Data leakage is defined as the accidental or intentional distribution of private organizational data to the unauthorized entities. A breach of security may lead to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data transmitted, stored or otherwise processed. Discussing and analyzing data leakage attacks and classifying them based on several aspects. There are several techniques currently in use in order to detect any defects or drawbacks which may be found in a cloud computing system. We aim at specifying and evaluating such techniques.

There are many types of data leakage problems that are currently known. We are working towards understanding and justifying the existing mechanisms used in order to mitigate the problem of data leakage. Despite the advanced mechanisms that are currently in use to mitigate the problems of data leakage, there are still several persistent problems that are not tackled efficiently by these mechanisms.We will study these problems and try to come up with any potential solutions which can be implemented in the future

OVERVIEW

Srinivasula Reddy Kalluri: The primary goal of this task is to explore different kinds of detection and defense mechanisms for sensitive data leaking in cloud computing. As the group leader, I was associated with sorting out and separating the work among different colleagues during the initial phase of the project. I assigned the work based on the interests and abilities of the team members and set the timetables for each individual part considering both the task project deadlines and their semester plan. The team successfully met all the deadlines with positive feedback from both the TA and the professor. The team was able to accomplish the tasks specified in the initial project report.

As a Team member, I was associated with analyzing and comparing varying types of defense mechanisms. In the early phase, I was engaged in studying about various challenges in Data Leakage Prevention Systems as mentioned in [1]. Further, I was likewise engaged in outlining different assessment measurements to survey differences in the areas of data breach identification and the introduction of Data Leakage Prevention systems. I contributed to section 3.3.4 in which the specific analysis of challenges in Data Leakage Prevention Systems. There are few challenges that we have to address while building a Data Leakage Prevention Systems. Detailed analysis and explanation can be found in the same section.

Rishti Gupta: As a deputy leader, I assisted the leader in conducting the executive tasks of the group including the minutes of the meeting, group meetings and motivating the team to work together to complete the required tasks on time. I have always taken the responsibility and have put in a constant amount of efforts to maintain the team spirit and work together to achieve the desired goal.

As a team member, I have researched various detection and defense mechanisms for the data leakage attacks in cloud computing for the project. In the early phase, I was involved in working on detection of the attack and discussions related to it as mentioned in [2]. I successfully completed the basic concepts of data leakage, its causes and different techniques to mitigate/ prevent the data leakage.

In the later phase, I worked with other team members on various defense mechanisms and to analyze which techniques lead to better results. I contributed to Section 3.2.1.1 in which specific approaches to defense mechanisms have been discussed. My research was related to the MetaData based Data Storage Model [24] which is an approach to make the data invaluable while the data is stored. The data is valuable only when the different fragments of data are mapped to each other. The observations and methodology can be found in the same section of the report.

Sai Hemanth Gantasala: I was assigned to work on a survey on data breach challenges on cloud computing[3]. This paper talks about the security aspects like confidentiality, integrity, availability and privacy issues of outsourcing the user data. The top security threats and challenges I have looked at are data breach, account or service traffic hijacking, insecure interfaces and APIs, Malicious insider attacks, economic costs of data breaches and shared technology vulnerabilities. I have also reviewed security related information-centric security, high-assurance remote server attestation, privacy-enhanced business intelligence and homomorphic encryption techniques. Also, I was also assigned to read about Mitigating Insider Data Theft Attacks in the Cloud [11]. This paper talks about a novel approach to protect the data using the combination of user behavior profiling i.e.., the data access pattern of a real user and decoy technology (i.e.., to place bogus information alongside the real information, so that an intruder is targeted to open a decoy file) to launch a disinformation attack on intruders.

Krishna Sumanth Gundluru: As a part of this project, I have worked on analyzing both detection and prevention mechanisms for data leakage and loss. In [5], the authors mainly discuss the usage of an open-source software MyDLP as a case study, which is used as a Data Leakage protection software to handle data loss and data leakage. It allows us to monitor, inspect and prevent all outgoing confidential data from the organization in use. It mainly focuses on content-blades which are highly accurate pattern matching detectors of sensitive data. [14] was about data leakage prevention and mitigation strategies in general. I split the work with Avinash, and was assigned the task of detailing the prevention strategies. Prevention strategies mainly discuss the different encryption methods on data before transferring it within the network.

Vishwarajsinh Sodha: I worked on analyzing defence mechanisms for Cloud computing systems. In [21], the author describes various multi-tier approaches to prevent data leakage. They go on to design an efficient Architecture and algorithm to prevent data leakage. Furthermore, [21] discusses in-depth various kinds of data leakages that can happen and how a robust algorithm can overcome all of those threats. They describe the architecture that can improve the response time as well as Security level compared to already existing architectures for security in Cloud Computing. Finally it describes the MULTI-tier approach that can help us have robust security where it describes various ways of securing OTP like using Festal Network Process and combining it with the first level of Security PIN as well. At the third level of Authentication they use Image Based Verification System that adds an additional level of security before the client is allowed to access the Cloud Server. Later, in [21] a lot of results and comparisons are drawn

with already existing and famous techniques for Securing Data Leakage in Cloud Computing Systems. It maps out and provides us with a lot of details that ensures us that the proposed algorithm is better in terms of security as well as in terms of the faster response time and serving the clients in a better way.

Nachiketa Pathak: For this project I have reviewed two research papers. I have worked on data leakage detection using cloud computing and various techniques that are used in detection of data leakage. I studied watermarking of data and various methods which are used for watermarking[6]. I also studied watermarking various types of data like media and relational databases. I also studied about the need for data allocation in cloud computing. I also studied how data can be secured from data distributors using k-Anonymity privacy protection. I have also studied about data warehousing and how to trace the data to its original source. I have studied various encryption algorithms that are used to secure user data in cloud computing. I have studied algorithms like Advanced Encryption Standard, Data Encryption Standard, the blowfish algorithm, RSA algorithm[20]. I have compared various characteristics of these algorithms: key size, memory usage, execution time, authentication type, data encryption capacity, scalability and security of these algorithms.

Akshay Agarwal: I have read 2 research papers as part of the project. I was responsible for analyzing a leakage detection scheme and understanding the change in trends in cloud security. The leakage detection technique I analyzed was based on watermarking. This technique in [7] explores using Discrete Wavelet Transform(DWT) as a specific scheme specially for image leakage detection. This scheme successfully tackles several attacks that an intruder can make in order to attack an image. [19] talks about the change in the mindset of people regarding cloud security. It starts from trying to make data insensitive to encrypting it to further making the cloud secure. It vaguely brings up schemes such as RSA, watermarking, firewalls along with their pros and cons.

Venkatraman Balasubramanian- I was assigned to work on the discussion of security in cloud computing and the impact of detection in cloud computing. To further delineate on the above, in [8] methods for detection of leakage in cloud computing is proposed. In this method a combination of two techniques is analyzed and its novelty expounded. Firstly, authors observe a security model that targets confidentiality based issues the Bell-La Padula security model. This model enables design and analyzes provisioning for securing computer systems. To this end, the model is popularly known as the data confidentiality model. Another model which works on data integrity is called the Biba-Integrity model. In this model the information system is divided in an object oriented approach. Major considerations in all of these are the length of the key, cipher, block size and analytical resistance of the crypts developed. Further security, prediction key and ASCII printable characters, time required for possible keys are all dependent on the

modelling. Specifically, DES follows a key distribution based approach and has certain key management limitations, however RSA has larger overheads in terms of time and encryption operation. Therefore in this proposed model authors provide a solution for the data leakage problem based on the Bell-LaPadula model for securing infrastructure and water marking the data.

Avinash Khatwani: For this project, I was assigned to review research papers to identify the ones that were within our scope of research. I started with understanding the whole structure of cloud computing and also the current risks involved with shifting to this technology. I worked on analyzing two research papers. The first research paper deals with Data Leakage Prevention Techniques[14]. This gave an overview of different types of data leakages, Mitigation techniques and also the steps a user can take to ensure data security of data in the cloud.

The second research paper discusses a model based approach for Map Reduce computational Security in cloud[9]. This method uses leading techniques to handle sensitive data. The three stages include Load Balancing, Map Reduce framework and Data Leakage Detection. The method proposed achieves good results by improving the overall performance and securing the data more effectively.

Pranjali Dileep Jagtap: In this project, I started the review of various research papers with my team members in order to shortlist a few which are relevant to our research. I studied the various cloud computing models and service models related to it. After knowing more about the data leakage in cloud computing, I got interested in knowing more about the detection techniques. I came across an interesting paper that talks about a “watermarking” algorithm that can detect if a data was leaked [10].Another paper describes a simple but elegant algorithm of “Tracing the data movement” in order to know the data movement and location at all times so a data leakage threat can be detected[26]. The paper describes the experiment, implementations and the algorithms in great detail and is a convincing method of data leakage detection.

DETAILED RESULTS

3.1 Discussion

3.1.1 Introduction

Cloud computing is an ideal way for accessing the computer system resources, mostly data storage and computing power. It provides a platform where active involvement of the user is not required. Many enterprises and companies are using this technology to store their crucial data. Since the data is highly confidential and important, its security is of the utmost importance. Thus, different models have been proposed in the cloud infrastructure to maintain the security of the data and prevent the data breachment.

There are mainly 3 types of models in [15] which cloud infrastructure is developed in:

1. Private cloud: This type of infrastructure is developed for an organization. Only a few specific and authorised users can access it.
2. Public cloud: In this case, the data on the cloud can be accessed by any public user from any location and at any time. The used may be charged as per the usage.
3. Hybrid cloud: This is a mix of both the above types. If a private cloud user needs to access a few things or services from a public cloud then the hybrid cloud can be used.

There are 3 types of cloud computing services:

1. Software as as Service (SaaS)
2. Platform as a Service (PaaS)
3. Infrastructure as a Service (IaaS)

One may wonder why the security of cloud systems is so essential according to [19]. Large businesses use cloud data centers to store some of their most sensitive information. If this information gets leaked or stolen, it can cause the organization unimaginable loss. Over the years, the methods to tackle data leakage have changed and evolved. Earlier, watermarking was considered as one of the main techniques used in order to detect data leakage. This mechanism has its pros and cons. Another thing a client would do was to make the data less sensitive before storing on a third-party cloud server. These days, many kinds of attacks and means have been developed by intruders. It was necessary that the means used to protect data from data leakage evolve. While the methods mentioned already use manipulation of data as a means to

secure data. Instead, it is important to make the cloud secure. Making the cloud secure will ensure that the data stored cannot be accessed by an intruder. For doing so, several schemes such as encrypting the data itself came into light. One of the most famous of them is the RSA Algorithm. While the encryption schemes focus on making the data secure, there are other schemes which can be used to make the entire network secure. Such a scheme includes setting up firewalls. Various schemes which encrypt data or perform watermarking may not be realistic. If the amount of data is very large, encrypting it or watermarking it may take a lot of resources and extra storage space. Therefore, it seems like a better and a more efficient idea to instead secure the cloud data centers. While RSA Algorithm successfully secures an image by encrypting it, the process can be very time consuming. If a cloud system works only on preventive measures, it will never find out if some intruder was able to bypass their security and access data. This form of being uninformed may lead to unknown effects on an organization. Therefore, it is equally important to ensure that a cloud system is able to detect when data has been leaked. Along with this, it needs to be ensured that fallback mechanisms are provided to take the correct actions to retrieve the data and track the intruder responsible for the leak. This will ensure that the organization is at least aware that an attack has taken place and data has been compromised. At the same time, they can ensure that the leakage does not cause them much inconvenience because of the fallback mechanisms in place.

3.1.2 Security aspects

The users may store a lot of personal and secured data on the cloud according to [15] which makes security very important. Following are the security issues in cloud computing:

1. **Confidentiality**: Preventing an unauthorised user from accessing an information is known as confidentiality. The users will always have apprehension about their confidentiality while uploading the data to the cloud.
2. **Integrity:** Preserving the correctness of data and also the consistency is known as integrity. Unauthorised person should not be able to change or modify the data on the cloud. Maintaining integrity is the job of the cloud provider.
3. **Availability:** When the system functions as it is expected to when needed is known as availability. The DDoS attack is an example where the availability of the system is compromised.
4. **Privacy Issues:** Since multiple users are storing the data simultaneously on the cloud, some privacy issues may arise. Some examples are loss of control, access control and invalid storage, data boundary etc.

3.1.3 Data Leakage

According to [2], data leakage is defined as the accidental or unintentional distribution of private or sensitive data to an unauthorized entity. The potential damage and adverse consequences of a data leak incident can be classified into the following two categories: direct and indirect loss.

- Direct Loss: It refers to tangible damage that is easy to measure and estimate quantitatively.
- Indirect Loss: Indirect loss, on the other hand, is much harder to quantify and has a much broader impact in terms of cost, place and time.

How was access to the data gained?

The "How was access to the data gained?" attribute extends the “Who caused the leak?” attribute[2]. These attributes are not interchangeable, but rather complementary. The classification by leakage channel is important for future reference and can be classified as physical or logical.

- Physical Leakage: Due to storage media is no longer operational.
- Logical Leakage: Due to deletion, partition corruption, partition deletion, format, reinstall, virus, etc.

3.1.4 Detecting challenges [2]

- **Encryption:** Data leaks in transit are hampered due to encryption and the high volume of electronic communications. While encryption provides means to ensure the confidentiality, authenticity, and integrity of the data, it also makes it difficult to identify the data leaks occurring over encrypted channels. Employing data leak prevention at the endpoint – outside the encrypted channel – has the potential to detect the leaks before the communication is encrypted.
- **Access control:** Access control provides the first line of defense in DLP. However, it does not have the proper level of granularity and may be outdated. While access control is suitable for data at rest, it is difficult to implement for data in transit and in use. In other words, once the data is retrieved from the repository, it is difficult to enforce access control. Furthermore, access control systems are not always configured with the least privilege principle in mind.
- **Semantic Gap in DLP:** DLP is a multifaceted problem. This approach lacks the semantics of the events being monitored. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching or access control scheme

cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios.

Existing System [2]

Data Leakage was handled by watermarking, e.g., a unique code is embedded in each distributed copy. Finding the copy in the hands of an unauthorized party helps to identify the leaker.

Proposed System [2]

The following algorithms are proposed to develop unobtrusive techniques for detecting leakage of a set of objects or records:

- **Evaluation of Explicit Data Request Algorithms:** Used to see whether fake objects in distributed data help to improve the detection of the guilty agent and to evaluate our e-optimal algorithm relative to random allocation.
- **Evaluation of Sample Data Request Algorithms:** With sample data requests agents are not interested in particular objects. Hence, object sharing is not explicitly defined by their requests. The distributor is "forced" to allocate certain objects to multiple agents only if the number of requested objects exceeds the number of objects in set T.

3.1.5 Modules [2]

- **Data Allocation Module:** Important to know how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.
- **Fake Object Module:** The distributor creates and adds fake objects to the data that he distributes to agents. Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents.
- **Optimization Module:** The Optimization Module is the distributor's data allocation to agents that has one constraint and one objective. The agent's constraint is to satisfy the distributor's requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. The objective is to be able to detect an agent who leaks any portion of the data.
- **Data Distributor Module:** A data distributor has given sensitive data to a set of supposedly trusted agents (3rd parties). Some of the data is leaked and found in an unauthorized place. The

distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means Admin is able to view which file is leaking and fake user's details also.

- **Agent Guilt Module:** This module is to determine fake agents. This module uses fake objects (which are stored in the database from the guilt model module) and determines the guilt agent along with the probability.

3.1.6 Types of Data Leakage

The different types of threats to data leakage as discussed in [14] include Internal and External Threats.

1. **Internal threats:** This results from insiders who get motivated by different means to send information to outside sources. As [14] suggests, the main reason for data leakage is due to internal threats i.e. up to 77% of the data security breaches. The internal threats can be further classified into:
    a. Intentional Data Leakage: This is due to outsourcing unauthorised data by the internal users. There can be various reasons for this such as financial gains, unfairness towards an employee etc.
    b. Unintentional Data Leakage: This occurs when an Authorised user sends confidential information to an unauthorised user by mistake. This is caused by mainly the carelessness of an employee and also unfortunate transaction measures.
2. **External threats:** This results from an outsider breaking into the security measures to hack the confidential information of an individual or an organisation:
    a. Data theft by intruders: This includes the intruders stealing sensitive information from visa card, MasterCard etc.
    b. SQL injection: These attacks mainly aim at stealing data from the database of the websites using SQL server in the backend.
    c. Malware: Various communications are initiated when a system gets infected with malware. This results in the loss of private data.
    d. Dumpster driving: It is important to destroy any sensitive data before disposing it into the dump where it would be recoverable.
    e. Phishing: Another common technique used by hackers to gain information of the users is by sending mails.

f. Physical burglary: Weak physical security can be exploited which may lead to loss of devices which contains sensitive information.

According to a survey, 80% of data leakage is seen in Personal information[14]. The other 20% information includes the commercial information such as organizational secrets, trading secrets etc. Both of these are damaging for a company.

3.1.7 Threats and Countermeasures of Cloud Computing

According to [18], to divide the security in cloud computing as based on "hardware, software and communication". Security issues in data centers are primarily due to lack of proper cryptographic techniques, additionally there are weak authentication measures which add to the aforementioned problems in cloud computing. In the following paragraphs we delineate the different categories of security measures in cloud computing:

1. **Embedded Security:** The ability of a system to connect to an external local network using high quality tools falls under this category. In cloud computing the major issue of this type happens in the usage of Virtual machines (VM). The isolation provided by these systems prevent stray user interference thereby providing strong security. However, deployment based issues can result in security breach. Further, workload deployment on these VMs can result in data leakage. Thus, the infrastructure providers should be careful when uploading such machines into the public domain. Moreover, controlling the VM to update the resource requirements and change host parameters need to be properly monitored in such environments.
2. **Application:** Software applications act as the bottleneck for security, the most sensitive areas that also act as points of vulnerability determine how robust the security is. Having both front end and back end security on any platform is essential. Large amounts of software codes need to be continuously monitored for security breach to not happen. Vulnerabilities arise due to bad maintenance of codes in such areas.
3. **Client Management:** Protection of client information in the public is known as client management. Client management or CM is an integral property of cloud computing security. Client experience plays a vital role in taking the needs of the cloud provider in terms of profit forward. As said in the literature, some providers fail to provide such solutions that leads to poor customer experience, which leads to poor security and thereby less profits. Authentication and

accounting as well plays key roles in the client management realm. This is the major determining factor for cloud provider selection among the customer groups.

4. **Cloud Data Storage:** Data storage plays a vital role in security of cloud computing data centers. All of the applications and physical devices need to be robustly put under a firewall for preventing security attacks. The growth in online applications makes it very important for the applications to be protected. Deployments of data warehouses posits the need for high security and shows the quality of a service provider.
5. **Cluster Computing:** A collection of multiple computers is known as a cluster. In cloud computing definition this means a collection of multiple Virtual Machines or servers looped together to maintain application execution in a parallel or serial fashion. In the industry, the parallel processing usage makes it necessary for having more robust security schemes to cater to the needs of multiple applications being executed at the same time.
6. **Operating System:** As said above, cluster computing involves multiple VM s this makes it important to check what are the Operating systems in use. As there are a diverse set of operation systems with multiple Vms, different servers, there is a possibility of having multiple OS. E.g. a VM cluster may have windows, linux, unix, or a different variety of such OS. Further, a networked system may have different OS types inside the routers which makes the heterogeneity more prominent. Security issues that are caused due to such heterogeneity need to be further analyzed and updated. These issues mainly arise due to installation limitations and manufacturer based policies. Further, in today's world, even cell phones have this heterogeneity which entices an attacker due to weak security properties. Smartphone based attacks are very prominent in the current research. Some of the issues like client side injection and destruction of native cryptography are some unsolved problems in cloud computing.

Further some of the other threats include, Data losses which can be posited as either " intentional or unintentional". These actions can be further classified as "both good and harmful intentions". Either of these have high chances of leading to data losses. On observation, it is clear that such losses can be due accidents or intentional alteration. The encrypted data can be lost by misplaced encryption keys etc.

There are losses due to disaster or Natural causes such as earthquakes or fires, but in cloud computing the threats are affecting the IaaS, PaaS, and SaaS services. The quality of a CC provider is determined by the way it ensures the data-loss aspect. These are put under the metrics such as " reliability, usability, and

extensibility". Although cost savings are an important aspect of cloud provisions, the methods should not compromise the data. In multi-tenant environments such as network slices and clusters, authorities need to be aware of the access methods being provided to the users. Incomplete authorization and lack of accounting can lead to issues in reliability and financial losses. It is very clear how data breaches can happen. For example, a breach can occur due to "incorrect authentication or authorization mechanisms, poor review of controls, undependable use of encryption keys, and operating-system failures". Companies like "Apple (iCloud), Microsoft, Yahoo, and Google" are a few cloud providers who have faced this threat. Majorly, encryption was the problem in these companies. Further, encrypting the data can lessen the abnormalities and effects of a data breach.

Moreover, third party intrusions are possible if the institutions are giving away their credentials to third parties for application developments etc. Malicious insiders can propagate falsified information that can lead to data insecurity or loss. This is because it is more difficult to detect a threat inside than outsiders. Malicious employees launch insider attacks on user sites, however, it has been observed that most security issues arise from insider attacks. For example, a malicious insider can gain passwords for important websites and launch an attack by hijacking. A noticeable case is of traffic hijacking which makes it very difficult to detect by an insider if the attacker is from the insider. Further, it is clear that most of the credentials of the cloud consumer are now available easily that makes it very easy to attack. Cloud account hijacking can be defined as follows "In cloud-account hijacking, a malicious intruder can use the stolen credentials to hijack the cloud services and then enter into others' transactions, add incorrect information, and divert users to illegal websites, causing legal issues for cloud service providers". As these threats are more prevalent today, the ability to obtain stolen information and pose it as an attack falls in the phishing and fraud schemes. These vulnerabilities are very difficult to detect in a cloud computing scenario.[18]

3.1.8 Data Leakage Mitigation Techniques

The following techniques have been discussed in [14]:

1. Source Content Management: This technique can be used to avoid the data leakage while using channels such as FTP, HTTP etc. There are various methods to achieve that such as
    - Detecting patterns, regular expressions done on gateway-based devices.
    - Detecting specific words performed by keyword filtering.

- Analysing data at rest and also creating databases with creation of fingerprints.

2. Reputation System: This system assigns a score for each user that sends out an email. This way unwanted emails from unwanted sources are restricted and hence phishing and spamming is reduced.
3. Thin Client: The users are strictly provided with only the necessary applications. This way disks and USB can be blocked which puts a restriction on copying data.
4. Markings based on Protection: In the method the sender of the email notifies the level of access needed to read the message. The recipients must clear the classification provided.
5. Web Filtering: Websites can be blocked according to the user's behaviour and history. This can restrict phishing sites.

3.2 Defense mechanisms in Data Leakage

3.2.1 Metadata Based Data Storage Model [24]

The data is being migrated to the cloud environment at a very fast pace these days, but the security of data predominantly remains a primary concern. Cloud requires security which depends and varies with respect to the deployment model that is used, the way by which it is delivered and the character it exhibits. Cloud computing moves the application software and databases to the large data centers, where the management of the data and services are not trustworthy, which poses many new security challenges. To ensure the security of the data stored in the data centers and other storage locations, the research paper proposes a new methodology which might not completely help in restricting a hacker to access the data but will make the data invaluable if it is extracted by a hacker but at the same time ensures the quality of the data that is being provided to its respective owner or authorized user. The methodology is a metadata-based data segregation and storage technique and also caters to how to access the segregated data. This methodology ensures that data is invaluable during static residence and gains value only during acquisition or while being updated. The solution will secure data during its time it is residing in the storage location. But again, this inherently triggers the need for designing ways to store and retrieve data. The research paper discusses the metadata based data storage model for the same:

According to this model[24], any information is of use only when the fragments of the information are available and are mapped/related to each other. For example, credit card number information without the corresponding information like CVV or credit cardholder name, etc is useless. Similarly, username and password are not valuable alone. The fragments of information become valuable only when they are

mapped together. In some scenarios, there is no need to store data in a mapped manner. But, mapping is required at the point of usage. Thus, a formal "proof of retrievability" (POR) model ensures the remote data integrity. It combines spot-checking and error-correcting code to ensure both possession and retrievability of files on archive service systems. Since the time of usage of information is less than the time the data is present at the storage location, two types of concerns arise:

1. During transmission (data usage)
2. During residing at storage centers (static phase of data)

To cater to the data transmission in the cloud, a layered framework to deliver security as a service in the cloud environment. This framework consists of a security service that provides a multi-tier security based on the need for the transaction. The framework provides dynamic security to users based on their security requirements, thus enabling a localized level of security and thereby reducing the cost of security for applications requiring less security and providing robust security to applications really in need of them. The model in the paper caters to the second problem that is data security at storage centers. This further has two concerns:

1. The actual physical unit where the data is stored.
2. The intrusion into the information.

The model mainly focuses on providing security in avoiding intrusion. Hackers can still get hold of the data in this model. It rather makes the data invaluable even if it is accessed by an intruder.

Design of the model:

Data has to be segregated into Public Data Segment (PDS) and Sensitive Data Segment (SDS). The SDS is in turn divided into smaller units until each fragment does not have any value individually. The fragmentation need not be of multiple levels. Instead, an effort has to be put in to identify the key element that makes the data sensitive and should be fragmented separately. Figure 1 demonstrates this data fragmentation:

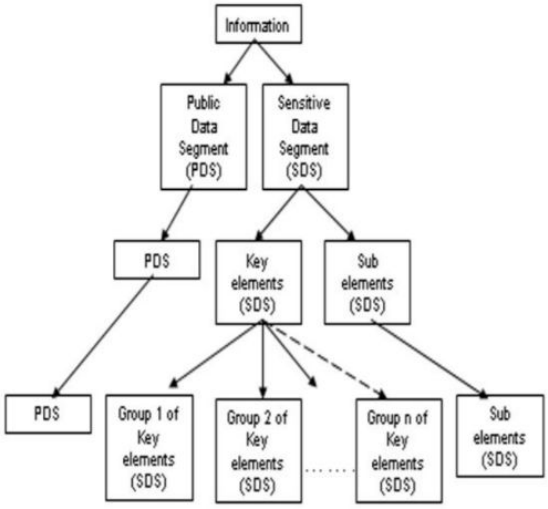

Figure 1. Data Fragmentation .[24]

The data is made invaluable in this process. But, mapping required to re-assemble the data should be conducted simultaneously while this fragmentation is done. As shown in Figure 1, This method is effective for a database that is being created from scratch and not for the enterprises that want to move their existing data to the cloud. For migrating data from the existing environment, the migration should be done appropriately. Thus, the model implements a Data Migration Environment (DME). The input to DME should be the existing schema of the database and additional information about the sensitive part of the schema should be given as Metadata to the DME. The DME can fragment the data into pieces based on the level of security needed. Parallelly, it constructs a mapping table to re-assemble the data.

Methodology:

The paper explains the methodology using the credit card example as described before: Consider a database in a bank containing information of the users with their credit card information. The schema for storing such information will be in the form of tables with some tables containing personal information of the user and some tables containing information regarding credit cards and will be mapped using their ids. This particular information can be stored in a database (let bankDb) as follows:

bankDB:

- A Customer table containing

    - CustomerId (Primary Key (PK)),
    - CustomerName,
    - CustomerAddress,
    - CustomerPhone,
    - CustomerDOB
- A Membership table containing
    - CustomerId (Primary & Foreign Key (FK))
    - Password, o PasswordQuestion,
    - PasswordAnswer
- A Creditcard table containing
    - CardId, (Primary Key)
    - CreditcardNo,
    - CardExpiryDate,
    - CVVNo
- A Customer_Creditcard table containing
    - CustomerId (Primary Key)
    - CardId (Primary Key)

An intruder can exploit this information if he gets access to this particular database as all the information is stored together in one place. The Customer table contains data that is not very important here. The membership table individually is of no value but if given access along with the Customer table, it is quite valuable for an intruder. The Creditcard table is sensitive data with high value because though there is no mapping done with the Customer table, it individually is a high potential target. For example, an online transaction can be done successfully with this data alone. If this table is given access along with the Customer table and Customer_Creditcard table, the bank will lose everything and become bankrupt. The entire data is usually stored in a single table and stored on the same hardware.

The model in the paper demands that the related data should be stored at different locations and should be mapped while updating or querying. The entire model is migrated to the proposed model through the DME. The user has to provide schema information to the DME and also supply the metadata.
The paper considers three categories of metadata in this example:

1. The data which is having low value is considered as ‘Normal’.

2. The data which is having high value is considered as ‘Critical’
3. The data which has value when mapped with other data is considered sensitive. And the data that maps ‘Sensitive’ or ‘Critical’ data to ‘Normal’ data is also considered ‘Sensitive’.

Table	**Metadata**
Customer	Normal
Membership	Sensitive
Credit Card	Critical
Customer_Creditcard	Sensitive

Table I: Metadata Information [24]

Now, the DME fragments this data. DME must be configured or customized according to the level of security needed. For instance, if medium level security is required from DME, the data in the ‘critical’ criteria should be fragmented. Similarly, if high-level security is required from DME, the data in both the ‘critical’ and ‘sensitive’ criteria should be fragmented. The DME is not aware of the actual data residing within these tables. Hence along with the metadata of the tables, the primary key column name should be provided in addition to it. This is easily available from the schema information of the database tables. The different levels of security needed and their corresponding metadata should be configured with the DME. The paper considers medium security for the data and uses the same for further implementation purposes. Thus, the DME can fragment only the data that is ‘Critical’.The dataset used in the example is from the ‘critical’ criteria. The corresponding table is Creditcard table and the primary key of this table is Credit card Id. As a first step, the DME fragments this table as below.

- DME_Creditcard table
    - SensitiveId (PK, Created by DME)
    - CreditcardNo
    - CardExpiryDate
- DME_Creditcard_Senstive table (Created by DME)
    - SensitiveId (PK, FK, Created by DME)
    - CVVNo
- DME_Creditcard_Mapper table (Created by DME)
    - Credit card Id (PK)

- SensitiveId (PK, Created by DME)

Table II lists the metadata of the database in which the data falls under the 'sensitive' category and depicts the current situation:

Table	**Metadata**
Customer	Normal
Membership	Sensitive
DME_Creditcard	Sensitive_DME
Customer_Creditcard	Sensitive
DME_Creditcard_Sensitive	Sensitive_DME
DME_Creditcard_Mapper	Sensitive_DME

Table II: Metadata Information After Fragmentation [24]

After fragmentation is completed, the DME segregates the schema, separating out the data modified by DME, 'Originally Sensitive' data and 'Normal' data is shown in Table III.

Normal	Originally Sensitive	Sensitive DME
Customer	Memebrship	DME_Creditcard
	Customer_Creditcard	DME_Creditcard_Sensitive
		DME_Creditcard_Mapper

Table III: Segregated Schema [24]

Then the 'Sensitive DME' data is then split into Actual Data (AD) and Mapper Data (MD).

- Sensitive DME
  - Actual Data
    - DME_Creditcard
    - DME_Creditcard_Sensitive
  - Mapper Data
    - DME_Creditcard_Mapper

The DME then segregates the 'Normal' data and the 'Originally Sensitive' data to different databases. It also moves the AD of 'Sensitive DME' data to a database at another location and MD of 'Sensitive DME' to the database with 'Normal' data. if the DME constructs its own table corresponding to the AD, then that table will be stored in a different location and will be the most sensitive. An extra mapping is required to map the original mapping table with the fragmented units. This is stored in a separate table. The database is of the following form:

Server 1 bankDB:

- Customer table containing
    - CustomerId (Primary Key (PK)),
    - CustomerName,
    - CustomerAddress,
    - CustomerPhone,
    - CustomerDOB

bankDB_DME

- Membership table containing
    - CustomerId (Primary & Foreign Key(FK))
    - Password,
    - PasswordQuestion,
    - PasswordAnswer
- Customer_Creditcard table containing
    - CustomerId (Primary Key)
    - CardId (Primary Key)
- DME_Creditcard_Mapper table containing
    - CreditcardId (PK)
    - SensitiveId (PK, Created by DME)
- DME_Mapper table containing
    - OriginalTableName (Combined PK)
    - NewTable Name (Combined PK)

Server 2

- DME_Creditcard table
    - SensitiveId (PK, Created by DME)
    - CreditcardNo

- CardExpiryDate

Server 3

- DME_Creditcard_Senstive table (Created by DME)
  - SensitiveId (PK, FK, Created by DME)
  - CVVNo

Original Table Name	**New Table Name**
Creditcard	DME_Creditcard
Creditcard	DME_Creditcard_Sensitive
Creditcard	DME_Creditcard_Mapper

Table IV: DME_Mapper Table [24]

Now each database contains data that does not have value in itself. An intruder who gets access to the data during the static phase throughout the data can not use the data to exploit the information in any way.

3.2.2 Mitigating Insider Data Theft Attacks in the Cloud

This paper is about a novel approach for securing data in the cloud using offensive decoy technology. By monitoring the data access and detecting the suspicious patterns in the data access, the suspicious user can be verified with challenging questions. So by launching a disinformation attack by returning large amounts of decoy information to the attacker, users data can be protected with high probability. The downside of startups and small businesses opting to outsource their data is that their data is prone to data theft. To make things worse, the customers have no means of detecting the unauthorized access of an intruder. The recent data theft on twitter is one such example where the intruder used twitter administrators password to gain access to twitter's corporate documents hosted on Google Docs. Even though much research has been focused on the security aspect of cloud computing, it is getting difficult to prevent unauthorized and illegitimate access to sensitive information. This paper talks about using a different security approach to secure the users data. It is called fog computing, where a numerous disinformation attacks are launched against malicious insiders and thereby preventing them by distinguishing real users' data to fake worthless data. The issue of providing secure access to users data remains the core problem of cloud computing and it is fair to say that the proposed solutions for security is failing due to numerous reasons like insider attacks, mis-configured services, faulty code and inefficient

implementations.

Methodology:

There are 2 ways to secure the user data in cloud computing.

1. **User Behaviour Profiling:** This is a common technique used in fraud detection systems, where the normal users behaviour is monitored i.e.., relevance and frequency of user access and thereby detect the abnormal user data access pattern.
2. **Decoy Technology:** Whenever an abnormal access is detected in the cloud service, decoy information such as dummy documents, honeypots and various other bogus information is generated on the fly based on the user profile and served to the intruder. In this way the real user information can be secured from unauthorized users.

Hence decoys fulfil two purposes namely,

1. Validating the data access if it is authorized or not
2. Confuse the intruder with useless data.

Hence the combination of these two security features purportedly improve the security of sensitive user data. So by combining user behaviour profiling and decoy technology it is viable to detect the illegitimate data access on a local file system by masqueraders (intruders who impersonate real users' data after stealing their credentials). A masquerader can be easily detected by observing the user behaviour profiling since the real user search is targeted or limited whereas the intruder is unlikely to be familiar with the contents and structure of the file system and hence his/her search is more likely to be wide spread. A machine learning model (support vector machine) is trained with one-class modeling technique based on the above stated key assumption. Also, the advantage with using this model is that the classifier has to train on user data which is isolated from other users and hence privacy and integrity of the user data is preserved. By simulating this model, all the masquerade attacks are detected with a very low false positive rate of 1.12%. Using the decoy technology, traps are placed within the highly-conspicuous locations of the filesystem and whenever a masquerader intrudes into the system, it is highly likely to access these files. Therefore monitoring the access to decoy files using Hash Message Authentication Code (HMAC) should signal the masquerading activity. HMAC uses the key which is unique to each user and uses it to compute the hash and stores it in the header of the decoy file.

The three advantages of storing the decoys in the file system are:

1. Detect the masquerading activity
2. Confuse the intruder and extra overhead to determine the dummy information over the real data
3. The amount of impact on intruders by serving bogus information, although it is hard to measure but it is significant in terms of preventing masquerading activity.

Furthermore an accidental opening of a decoy file can be ignored if all other data access patterns are deemed to be normal. In sum, detecting abnormal data access patterns and decoy traps can make an effective masquerade detection system and also improves the detection accuracy. The decoy files are generated on demand based on the user data such as tax return forms, medical records, ebay receipts and credit card statements e.t.c. To check the correctness of this approach 18 users data is collected over a period of 4 days and a classifier is trained for respective data using anomaly detection detection of search behaviour [26]. These classifiers are tested using simulated masqueraders and the results (Figure 2) show that this approach is better than the behavioural search profiling approach alone.

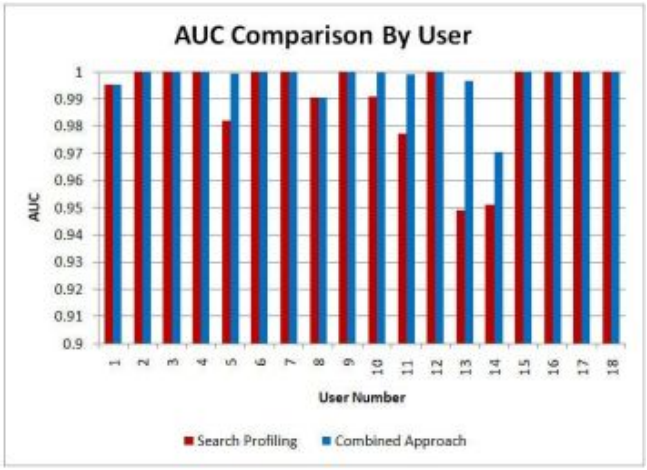

Figure 2. AUC comparison by user model for the search profiling and integrated approaches [11]

Hence the test results observed in Figure 2 suggests that as the user number increases the AUC ration fluctuates in an arbitrary manner. It is sufficient enough to show that user profiles are accurate to detect the unauthorized access. Hence this approach can be adopted in cloud environments to monitor and protect the real user's data from an intruder.

3.2.3 Efficient architecture and algorithm to prevent data leakage in Cloud Computing using multi-tier security approach in [21]

Existing Techniques

In [21] authors consider the existing techniques for Data security and propose new techniques with improved Response time and more robust security. Cloud computing offers various services like software, hardware, data storage, platform, infrastructure and many more. Cloud is highly accessible over the internet, which imposes major threat issues to it. Hence we need strong authentication mechanisms for better security. In [21] there are several procedures in place to avoid these attacks. One of the most commonly used measures is OTP. It is heavily used to do some online transactions through M-Banking , E-banking. This makes it extremely important to make our OTP secure from attacks like phishing, man-in-the-middle attack, malware trojans, internet hacking or mobile thefts. This paper refines the encryption method of OTP along with usage of top secret PIN to provide proper authentication and security. The paper discusses Festal Network Process that can be used to secure OTP. By using this the size of the input provided can be easily changed. The sub-keys are generated in each round, so the more the round of encryptions harder it becomes to crack the OTP.

Proposed Work

In [21], further discusses some advanced challenges that may occur such as the cases of unforecasted Data leakage happening because of unexpected misconfiguration or presence of virus in cloud platform. Here a Cloud Safety Net (CSN) a prominent monitoring framework that gives tenant visibility into the spread of their request data in a cloud environment with low performance overhead. Using a tag based approach it monitors data flows based on the approaches like Flow management Techniques, encryption and digital watermarking. With the help of this we can detect Data Leakage and prevent data from going into wrong hands. As a part of the process, online verification is conducted and an Electronic ID (E-ID) is generated.

[21] also discusses the list of parameters that influence the security of the cloud. For example one of the important functions a framework can perform would be to develop a way to monitor cloud management's software or monitor the application of the client itself. Finally one can also monitor closely the actions of the client and check whether he allows the automated patching software to run, or lets update software antivirus definitions, or check whether they have an understanding of how to robustize virtual machines in the cloud. By ensuring the parameters listed by the author are fulfilled we can make sure that data is in the right set of hands. The paper also goes on to discuss multi dimensional password generation algorithms at

various levels. Furthermore, the proposed algorithm makes response time even faster as it uses less bandwidth and lesser amount of resources as compared to existing techniques.

Breakdown of Security Components: Following are the different components for the security measures in [21]:-

Secure KeyBoard: This analog device is exclusively designed to allow isolation between connected computers. With the advent of the Internet, Computers are usually connected to diverse networks so isolation among these networks is necessary to avoid data leakages. SYSTEM keyboard is disable when SECURE keyboard is in use. Logging in through a security keyboard is mandatory when users want to access the cloud services.

Login Process: If by any chance the user is unable to login through the above process, then an alert message is sent to the client's registered number.

OTP: For a single session on a computer system or any other electronic device on the cloud, we generate a random numeric string and send it to the user. It is a static password and it won't be vulnerable to replay attack, as it is not valid for a long period of time.

Image Based Verification System: Herein Images are displayed in a randomly shuffled manner and the user has to choose the password to successfully pass through the Image based verification system.

Proposed Architecture

Let's understand the different components of the Proposed Architecture in [21]. As shown in Figure 3, we have :-

1) USER

   As the first step, the user sends a request to the service provider serving the master server −1 to access the services. Then the user has to fill in the register by filling the mandatory form via active email-id. This form has been sent by a cloud service provider to the client. The numeric code has to be provided by the user at the time of registration. Once the registration is done it will act as a first level security code for the user. The code also remains static.

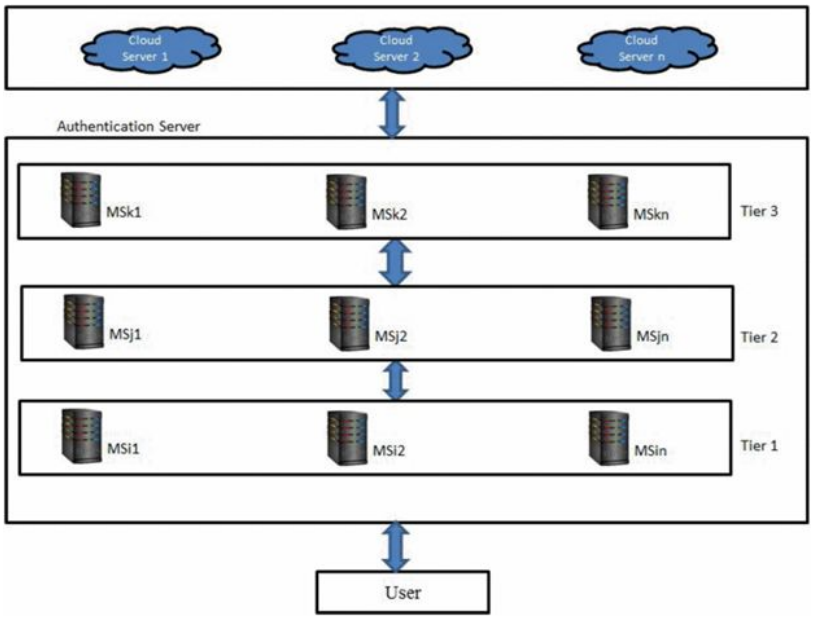

Figure 3. Proposed Architecture [21]

2) MS-1

The master server-1 stores all registration information of clients in encrypted form into their database. As the user enters the security code for first level authentication via secure keyboard, the password is encrypted and matched with the stored code. If it matches the user is redirected to Master Server-2, otherwise to master server-3

3) MS - 2

As we know, the failure at MS-1 the user is redirected here. Here the OTP is sent to the registered mobile number of the Client. The OTP stays valid for a limited time. If the authentication via OTP is successful at this second authentication level strata, the user is redirected to Master Server-3 for further processing

4) MS - 3

Here, the user has already passed the first and second level of security authentication. Here image based verification system launches random image security code. After the successful authentication the user is allowed to access the cloud services. Once all 3 levels of authentication is done, Master server-3 takes a decision of choosing the nearest available server for the client to fulfill his cloud computing needs.

5) Authentication

This layer provides an interface to cloud services users to login and authenticate themselves to be able to successfully connect with the Cloud services provider. This is used to check the authenticity of the users at multiple levels and is present at master server.

6) Cloud Server

Finally, after all of the authentication and authorization the user is allowed to access the cloud server. Here he gets to access the different kinds of services like SaaS, PaaS and IaaS. It is a kind of virtual server built and hosted over the internet through cloud computing platforms.

Proposed Algorithm in [21]

Multilayer authentication (NSC: Alphanumeric security code, ENSC: Encrypted Numeric Security Code, RISC: Random Image Security Code, DB: Database, OTP: One Time Password, MS1: Master Server-1, MS2: Master Server-2 MS3: Master Server-3 RMN-Register Mobile Number) in [21]

- Step 1: Registration for new users redirect users to NSC and ISC chosen by the administrator of the organization an encrypted numeric code stored in DB. [21]
- Step 2: Numeric User ID and Security code is entered by a special keyboard for the level authentication and it must match with ENSC stored in MS1. [21] If Match is found, then First level authentication is successful and goes to step Else Match not found, then OTP has been sent to the RMN.
- Step 3: OTP at MS2 and sent to the RMN
- Step 4: OTP has to be entered by the administrator.
- Step 5: Generate new security code at MS2 and send it to the RMN
- Step 6: New Password has to be entered for authentication. If match found then Second level is successful
- Step 7: Third level random ISC selected at MS3 by the administrator of the organization stored in the DB.If Match found, then the user permitted to use Cloud Services

  Else Match not found, then Go to step 3. [21]

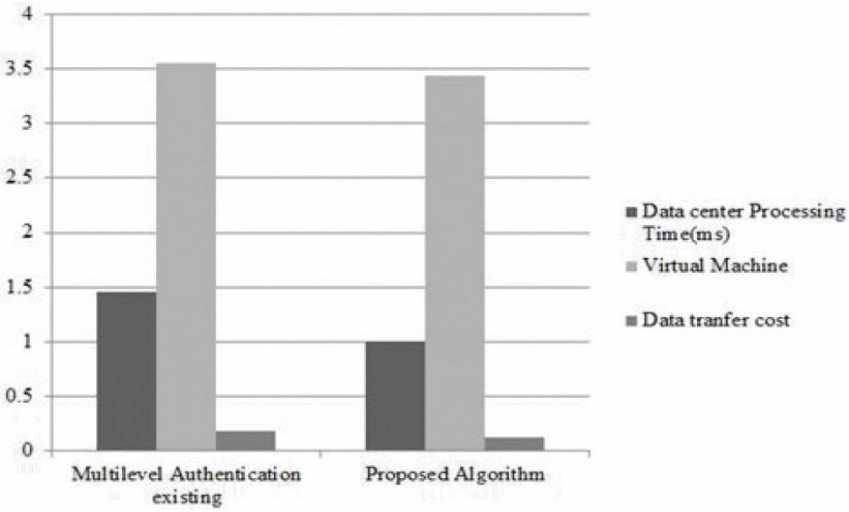

Figure 4. Comparison of DC Processing time, Virtual Machine and Data Transfer cost for existing and proposed algorithm [21]

Below Figure 5. draws the comparisons between already existing Multilevel Authentication and the proposed algorithm over the factors like Overall Response Time in [21]

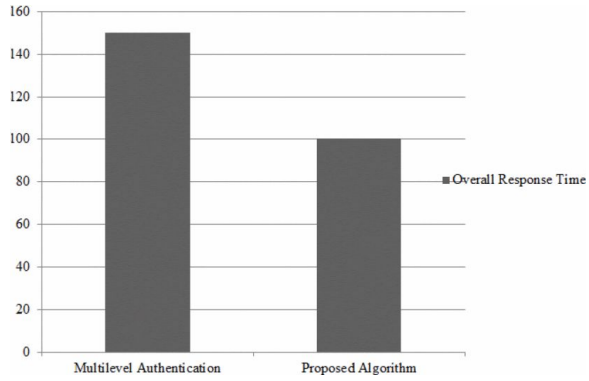

Figure 5. Comparison of Overall Response Time for existing and proposed algorithm [21]

Results and Comparison

Following are the result comparisons with the statistics of already existing techniques in [21]. Below Figure 4 draws the comparisons between already existing Multilevel Authentication and the proposed algorithm over the factors like Data center Processing time, virtual machine and data transfer cost in [21].

Below Table V. shows the overall statistics that are gathered from comparing different factors in [21].

Technique	Overall Response Time (ms)	Data Center Processing Time (ms)	Total Virtual Machine Cost (USD)	Total Data transfer Cost(USD)
Multilevel Authentication Algorithm (EXISTING)	150.3	1.46	3.55	0.18
Proposed Algorithm	100.24	1	3.44	0.12

Table V: Performance Measures Comparison between existing and proposed algorithm [21]

3.3 Detection mechanisms in Data Leakage

3.3.1 Watermarking

Introduction:

In the modern world where we use the internet for everything including bank transactions to filing for taxes, a lot of sensitive data exchange takes place over the internet. The companies handling this data are responsible to keep this data secure. The data needs to be protected from the cyber criminals who might steal some sensitive information and abuse it. Data leakage happens when any sensitive information about a client, code or any design specification gets into hands of a cyber criminal. The cyber criminals use the leaked data for their own profits causing losses to the company. The authors have tried to solve this problem by suggesting a novel technique of adding fake objects to the data that needs to be distributed in order to find the cyber criminals who misuse the data.

Watermarking the information:

The security technique of embedding a code or encryption on the distributary data information is known as the watermarking technique[10]. A company can claim ownership based on this encryption. The information can be in the form of text, image, video or any kind of file. In this technique, tuples and subsets of the data are identified and encrypted with a company controlled key. A bit pattern is added at a particular position on this tuple and subset.

Digital watermarking schemes can be used in order to detect if a data leak has taken place in the cloud. When making copies of data in the cloud, it is important to ensure that these copies remain secure and none of them have been leaked to an intruder. Digital watermarking schemes allow invisible watermarks to be present in the data such that they can be used in order to trace back to the rightful owner of the data. In case a leak has taken place, it can be detected by extracting the hidden watermark in the data.

The original data or the watermark pattern is not accessible. A small subset of data with a portion of watermark can be used to detect the watermark. The watermarking software feeds the information with small errors. This combination of errors and information doesn't make a significant sense and hence cannot be abused by any third-party intercepter. If any digital data like images, videos or documents gets leaked, this problem can get mitigated by watermarking since it can act as a proof of ownership.

Need for data allocation:

Information system is a process in which a company allows another company to access its sensitive information. This data can be very sensitive and hence needs high level security in place. Security issues can range from monitoring that the software and other data is being used by the right users, data privacy, data gathering and distribution according to legal and political guidelines etc. If an organization sends some information to the trusted third party and if the party breaks the trust and leaks the information, then the data is said to be open. When the distributor identifies the presence of a set of data objects at a new location that is a data leakage, then the main task is to identify the location of the agent who leaked the data. The next step is to identify the agent and to stop them from accessing any more data.

In order to increase the probability of locating the agent who leaked the data, the distributer sends the data using data allocation strategies and adding fake objects in the data. Assumption is that the agent would again leak the data. Now depending upon what fake information is out on the internet, the distributor can know which person leaked the data. Enough evidence is then gathered against the agent and legal action is taken.

Cloud Computing data leakage:

A cloud is an interconnection of multiple virtual systems. These interconnected systems and the data present on them can either be private or public depending upon the specific user needs. For example, an example of cloud computing is the OneDrive by Microsoft [10]. A user can access the data on their

OneDrive using their private user id and password. The data resides in the cloud and can be accessed from anywhere using the internet with a valid user id and password.

The data present on the cloud can open up wide access throughout the world. Any person can access the sensitive data using an authorised id from anywhere in the world. The cloud infrastructure stores the data virtually by using a cloud server. The cloud server is maintained by using HTML,XML code. There is no physical data when using cloud computing. According to Google, some of the properties of cloud computing are [10]: User-centric , Task-centric, Powerful, Accessible, Intelligent & Programmable

Data Leakage Detection Using DWT:

It is essential that the watermarking scheme used is robust towards data modification and manipulation. One such scheme mentioned by [7] in order to secure image data is watermarking an image using Discrete Wavelet Transform (DWT). Discrete Wavelet Transform splits an image into 4 sub bands. These sub bands are formed by passing an image through a pair of filters. Passing the image twice through the low pass filters creates the LL band. Low pass filters followed by the high pass filters forms the LH band. High pass filter followed by the low pass filter generated the HL band. Finally, the HH band is formed through a pair of high pass filters. The HH sub band tends to have more high pixel intensity values. On the other hand, the LL subband is approximately the original image itself. Multiple levels of this transform can be applied on the image by reiterating the same process on the most recently generated LL sub band.

The proposed scheme performs 3 levels of DWT on the image which is to be watermarked. The owner of the image can then choose an image which will be used to watermark the original image. Next, 3 levels of DWT are performed on this image as well. Let us refer to the final LL subband created for the original image as LL3. Let us refer to the final sub band of the image to be used for watermarking as wLL3. Using any value ‘b’, we can modify the sub band LL3 as:

$$nLL3 = LL3 + b*wLL3$$

Using the new modified sub band nLL3 instead of LL3 and the other sub bands of the original image, the image is reconstructed by performing 3 level inverse DWT on the sub bands. Using the correct value for ‘b’ will ensure that the watermark is completely invisible for the newly watermarked image.

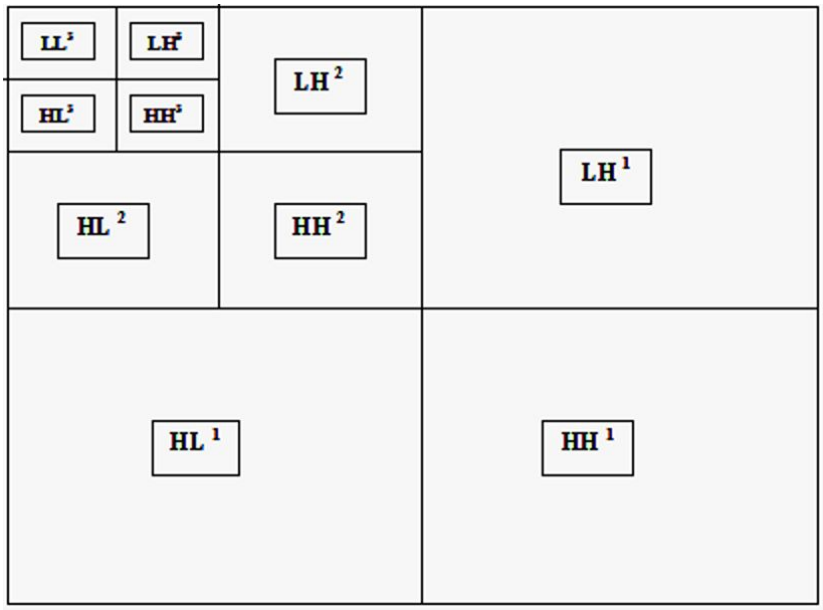

Figure 6. Level 3 DWT of an image [7]

The only way the watermark can be removed from the new image is if access to the original image is there. As shown in Figure 6, Without the original image, an intruder will not be able to remove the watermark through any form of attack or modification. In order to extract the watermark, the following formula can be applied:

$$wLL3=nLL3-LL3/b$$

Each of these values can be attained by performing a 3 level DWT on the original image, the new watermarked image, and the image used to perform the watermarking.

3.3.2 Data leakage detection and prevention using MyDLP[5]

Cloud computing in recent years has seen a great boom in its usage due to the variety of advantages it provides such as pay per usage of computing resources, no maintenance costs of infrastructure, etc. But as any technology has its risks, cloud computing has its own. One of the major concerns with cloud computing is data leakage. Data leakage is defined as the unauthorized exchange or transfer of critical information from one party to another which are using the same computing resources. Data loss/leakage protection (DLP) is one of the effective methods in dealing with data loss. DLP is a way to identify, monitor and protect data in use, motion and rest. DLP is used to identify sensitive content with the use of deep content analysis and enforces protective controls to prevent unwanted incidents.

This paper is qualitative research which is developed as a case study focusing on analyzing DLP methods in minimizing data loss and data leakage problems in conjunction with previously used technologies. [5]

Data Collection Methodology:

Main source of data used for this case study has been from interviews, observations, documents, and reports of a company's internal knowledge base (real time data or empirical data) and conducting interview questions to gather the project details. Both closed and open- ended questions were used during the interviews, and the interviews were performed in an email system. In addition, security journals, DLP books such as (Data Leak Prevention - ISACA) were also used.

Types of Data leakage:

Three types of leakage have been discussed, mainly unintentional, intentional and malicious leaks. Unintentional leaks are described as leaks which take place when a user mistakenly transfers confidential data to a wrong recipient or a third party. Intentional leaks are described as leaks which take place when a user transfers confidential data to a wrong recipient or a third party in ignorance such as lack of awareness of the company's policy. Malicious leaks are described as leaks which take place when a user deliberately transfers confidential data to a wrong recipient or a third party.

Data leakage prevention using MyDLP:

MyDLP is an open source data loss prevention software that runs with multi-site configurations on network servers and endpoint computers. It allows us to monitor, inspect and prevent all outgoing confidential data from the organization in use.

The various features it includes are as follows:

1. Blocks outgoing confidential data from the organization's network in use through both mail and web. It also archives files that have been detected as suspicious.
2. All portable devices such as USB and mobile phones usage is monitored, and transfer of confidential files are blocked when attempted.
3. Printers access to confidential data is also blocked.
4. Keeps track of confidential data at all times on network storages, databases, workstations and laptops.

These are few essential features which are key in protecting data leakage.

The software has a dashboard GUI which is updated at all times. It categorizes incidents by network, endpoint and datacenter and shows the status and number of incidents generated at each point of time. Automatic notifications are generated to alert in case of incident outbreak.

Content-blades are an important aspect here. These are highly accurate pattern matching detectors of sensitive data. It is of two types. Described-content blades are defined as detailed descriptions of sensitive content which contain terms, regular expressions, programmatic entities that enable accurate detection of classes of sensitive content. This information for an example may be Social Security Numbers, etc. Fingerprinted-content blades are defined as mathematical descriptors of individual sensitive documents or fragments of documents. These features are used to match copies of any documents or fragments. Fingerprints of known sensitive documents are created which are later used to ensure that unauthorized copies of the documents are not being used. The DLP products use content blades to perform content analysis on intercepted messages, stored files, and files being manipulated by users. Each document or message is assigned a score, or risk factor, depending on how strongly it matches a content blade. [5]

Alerts to these detected files are also semi-automated through policies. Policies are sets of rules that specify when to create an event (a record that a sensitive document or message has been detected) and how to act on, or remediate, that event. A policy can base its decision on the results of content analysis (the risk factor, or severity, of the analyzed content) and on non-content-based factors such as the identity of the message sender or the destination of the user action. There is also a dedicated workflow followed to analyze the root cause and follow the remediation process. A watch list is maintained on all the user activities. Warnings are generated if the security incidents are repeated, which involve reporting to other departments like legal, compliance etc.

3.3.3 Data leakage detection - Dynamic Data Leakage Detection model-based approach for MapReduce Computational Security in Cloud

The method proposed[9] deals with the growing concern of security in the world of cloud computing. This method uses the leading techniques to handle sensitive data. The architecture developed can be used for fast and efficient processing of large datasets. This structure provides reduced data which leads to recognition of Guilty Agents that discloses sensitive information. The proposed methodology deals with three stages i.e. Load Balancing, Map Reduce and Data Leakage Detection. **Load Balancing** assists with the distribution of load among various resources to achieve minimum response time and maximum

utilization of resources. The main motivation of using load balancing in this paper is to accomplish better performance and to maintain sustainability. The purpose of using the **Map Reduce framework** is for analysis of filtering and reducing the dataset. Hadoop has been utilized for the execution of Map Reduce Framework. Data Leakage is the process of unintentionally distributing sensitive data to unauthorised destinations with the goal of assessing guilty agents as a probability of having a part in the leakage.

The Proposed Framework[9]:

Considering Di as the input data that is stored in one of the cloud servers and D in the range of D = D1 U D2 U … Dn., the load balancer is assigned with Dn to manage the entire load. This is followed by reducing the load at each appliance using a Map Reduce Framework which results in a reduced version of the data (Dn). After which Distributors are used to make the data more secure and helps in finding the guilt agent in case a data leakage takes place.The method discussed deals with distributing equal Workload to each Appliance by using load balancing and map reduce approach. The output of load balancing is put into mapper instances which helps in the distribution of data according to the filtration of similar types of data by storing them in terms of key value pairs. Following which the reducer function takes the intermediate keys and combines the values to create a small set of tuples. A security layer is implemented on top of the reduced data to detect any kind of leakage and by using Probability methods, the responsible Agent for the leakage can be identified.

Flow of Work:

Load Balancing: This component's responsibility is to achieve dynamic balance of workload, ideal utilization of virtual machines, reducing the waiting time for resources on the cloud, and minimizing the consumption of cloud infrastructure. Various parameters come into the picture when dealing with Load Balancing such as Total Bandwidth, Transfer Rate, Total Data Transmission, Transmission delay etc.

Round Robin Algorithm has been implemented for Load Balancing. It works on the basic principle of allocating an equal amount of time on Virtual Machine for the incoming requests which are stored in a queue. The processes are switched in a circular order without assigning any priority in which the incomplete jobs are resumed in the next cycle. The main advantage this algorithm provides is that it utilizes all the resources in a balanced order and also helps to reduce the starvation problem.

Map Reduce Model: After the whole data has been balanced to the Load balancing component, the Hadoop File Distributed File system assigns the workload to a mapper instance according to the filtration of the same type of data. It improves the efficiency by implementing parallel processing of large data sets.

Hadoop maintains large datasets in two phases i.e. Mapping and Reducing. The system consists of Master servers and slaves. There are multiple slave machines i.e. task trackers whose job is to perform computations and report to job trackers. The master looks over choosing and allocating members to execute the map and reduce functions.

The overall functionality of the Map Reduce framework is dependent on the <key, value> pairs. This groups the intermediate values to the same intermediate key and passes them to the reduce function. The reduce function works with the corresponding values and returns a smaller set of tuples. The map reduce model provides high level penalization.

Securing Map Reduce Components: This is the most important component while dealing with securing the data. The reduced data is secured by detecting Guilty Agents (GA). Security alarms are implemented to alert a user in case of a data leakage. The reduced data is distributed among the corresponding distributors, these distributors (DIi) pass the data to Agents. The advantage of using this system is that it does not alter the original data unlike the previously used watermark technique which inserts a code onto each copy. The purpose of data detection is basically to find the guilty agent i.e. the Distributor DIi finds the responsible Agent Ai. The challenge here is to recognize the Guilty Agent when the distributors confidential information has been. S-max Algorithm has been found to be the best among the others to find the probability of identifying the guilty agent. Further the focus lies in minimizing the overlap between the distributors while allocating the information. The s-max algorithm is applied to the reduced data which helps in assigning the objects acquisitively to optimize maximum objective. This results in minimum increase of the maximum relative overlap between any two given agents.

Results and analysis:

This paper experiments with the weather forecasting data accumulated from an Indian government website. The results show the assignment of equal quantities of data to each virtual machine and that the virtual machines have approximately the same response time. The framework also reduces the data by 70-80%. It has also been noted that the probability of finding guilty agents decreases as the number of agents increases i.e. it is inversely proportional.

Overall, this paper focuses on developing a secure architecture by analysing the compositional structure of the cloud. The method proposed improves the overall performance effectively and the data more secured by avoiding data leakage. The working is formed by mainly Load Balancing, Map Reduce

Framework and data leakage detection. Very good results are achieved. The future work can be extended by incorporating the trust on third party agents which would further enhance this system against insider threats.

3.3.4 Specific analysis of challenges in data leakage prevention systems

Leaking channels: Private information can be spilled through channels, for example, USB ports, CD drives, web benefits and printed reports as shown in below figure. We can force intelligent access rights on the off chance that somebody is getting touchy information, yet similar information can be accessible in types of printable reports. Web administrations and record sharing are a portion of the spillage channels that are related with information spillage in the cloud as described in [1].

Human Factor: Human activities are affected by numerous mental and social factors, that is the explanation it's in every case hard to estimate human conduct. It includes the subjectivity in taking activities, for example, giving access rights characterizing the degree of mystery to information as mentioned in [1].

Access Rights: It is constantly significant for information spillage frameworks to recognize the clients dependent on privilege rights they have. Information Leakage Prevention Systems will have a predefined set of clients list who approaches the records. On the off chance that it doesn't have the rundown, it is extremely hard for the frameworks to separate the clients. Simultaneously out of date get to rights may cause information spillage. For instance, If we don't repudiate the entrance privileges of any downsized or expelled workers, quite possibly they may get to information utilizing their old access in [1].

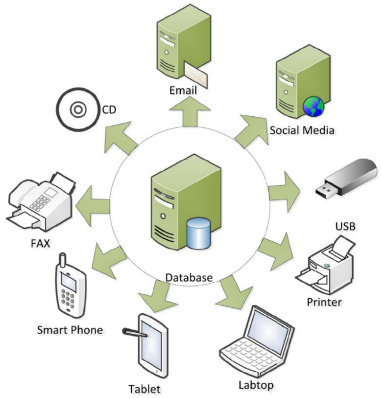

Figure 7. Different channels for leakage[1]

As shown in Figure 7 the channels of leakage cannot be restricted to just one source. We further elaborate this in the following sections.

Encryption and Steganography: System based DLPS typically attempt to distinguish duplicates of the delicate information utilizing distinctive examination techniques and attempt to contrast and unique information. For instance a delicate document is scrambled and attempted to send it as email connection, DLPS attempts to check the encoded record with unique information, however it won't identify the scrambled document as touchy record spillage right now in [1].

Data Modification: The recognition takes puts as a rule when examples or closeness with high rate is taken note. As a rule in the greater part of the information spillages, touchy information won't send all things considered. They may include or subtract a number of lines or they may include sections. They may change the report semantics by composing a detailed clarification of the substance which by and large makes the record as various one and it wont get recognized by the framework.

A few Data Leakage Prevention frameworks use hashing techniques to check the information spillage. Hash esteems and existing touchy information will be contrasted with check for comparability. On the off chance that the two qualities are coordinated, it implies an information spillage is recognized. The main issue with these kinds of frameworks is that on the off chance that you adjust a slight line it prompts a change in hash esteem. The better methodology is to isolate the information into sections and afterward ascertain hash esteems for those littler parts as described in [1].

Data Classification: Data Leakage prevention systems usually depend on data classification. If the data is not classified properly into different levels, these systems will not be able to distinguish between sensitive and normal data. For example, In the military they use confidential terms such as classified, restricted, top-secret words, This makes the system more oriented to protect one type of data in [1].

3.3.5. Data Leakage Detection using Tracing algorithms

Overview:

In cloud computing, detecting data leakage can lead to a great security risk. Hence, it becomes very important for all stakeholders to be able to track their data on the cloud. This can be done with the help of event management and also needs strong transparency and accountability.

In order to facilitate the data tracing in cloud, following things are needed[27]:

1. A logging mechanism that has the records and tracks of data lifecycles and movements in the cloud.
2. A sophisticated and powerful SIEM tool (Security Information and event management) that can take data-events as input, process it so as to correlate the input with other data events in the cloud in order to reduce false positives.

Common Atomic actions:

The tracing algorithm in this paper uses a five layer framework. These layers are [27]: Systems - track the data events at file and block level, Data - analyse the data logs, Workflows, Laws and regulations, Policy

Rule-Based Data Provenance Tracing Algorithms[27]

There are two main domains where data leakage can happen:1. Same machine (local) or 2. Across machines.

A. Data provenance Tracing algorithm for local machine

The aim of the tracing algorithm is to detect those file operations that can leak content or grant access to an unauthorised user. These file operations can implicitly indicate a data leakage. In Linux, the operations are file copying, renaming and file movement [26].

As shown in Figure 8, **In algorithm 1**, File copying is detected i.e. if a new file was created which had similar contents to the existing file. This can lead to data leakage because ownership of the new file can be different and full control may be granted to an unauthorized person.

As shown in Figure 9, **In algorithm 2,** File renaming is detected i.e. if the name of the file is changed while the content remains the same. This might lead to the old file in a temporary location due to auto-save function.

As shown in Figure 10, **In algorithm 3**, File movement is detected i.e. if an existing file is moved to a different directory. This may be dangerous if the file is moved to a removable location.

B. Data Provenance Tracing algorithm for Cross-machine Data Leakage Problem

In the local data leakage problem, the file operations are atomic and run in a single process and hence easier to detect. But in case of cross-machine file transfers, they usually use up multiple processes. Therefore, in this case we cannot design the algorithm by detecting just the action patterns that are executed by multiple processes. Hence, in this case, processes are organized in

the form of tree structure and detect actions in a process tree. Usually, cross machine transfers prone to data leakage are copying files over machines or email clients.

The authors experimented with the 'scp' program and the mail client.

As shown in Figure 11 and 12, **In algorithm 4 and 5**, SCP on sender side and on receiver side the threat is detected by occurence of related events like open action by scp and connect to inet by ssh, which if executed together, can be used to detect a file copy action.

As shown in Figure 13, **In algorithm 6,** the file attachment detects a threat that a file gets transferred in the email attachment.

Algorithm 1 Algorithm Detecting File Copying

```
for each process p in the OS do
    if p executed both Read and Create
    and Read was executed before Create
    and the file been Read is not the file been Created
    then
        return "A file has been copied!"
    end if
end for
```

Figure 8: Tracing algorithm 1 pseudocode [27]

Algorithm 2 Algorithm Detecting File Renaming

```
for each process p in the OS do
    if p executed both Rename (Old File) and Rename
    (New File)
    and Rename (Old File) was executed before Re-
    name (New File)
    and the file been Renamed is not a temporary file
    and the file was in the same path before and after
    Rename then
        return "A file has been renamed!"
    end if
end for
```

Figure 9: Tracing algorithm 2 pseudocode [27]

Algorithm 3 Algorithm Detecting File Movement

```
for each process p in the OS do
    if p executed both Rename (Old File) and Rename
    (New File)
    and Rename (Old File) was executed before Re-
    name (New File)
    and the file name was unchanged before and after
    Rename
    and the file path was changed before and after Re-
    name then
        return "A file has been moved!"
    end if
end for
```

Figure 10: Tracing algorithm 3 pseudocode [27]

Algorithm 4 Algorithm Detecting File Sending

```
for each process tree t in the OS do
    if two processes p1 and p2 belong to the process tree
    t
    and p1 executed a connect_to_INET syscall issued
    by ssh
    and p2 executed an open syscall issued by scp then
        return "A file has been sent using scp!"
    end if
end for
```

Figure 11: Tracing algorithm 4 pseudocode [27]

Algorithm 5 Algorithm Detecting File Receiving

1: **for** each process tree t in the OS **do**
2: **if** two processes p_1 and p_2 belong to the process tree t
and p_1 executed a **connect_from_INET** syscall issued by sshd
and p_2 executed an **open_new** syscall issued by scp
then
3: **return** "A file has been received using scp!"
4: **end if**
5: **end for**

Figure 12: Tracing algorithm 5 pseudocode [27]

Algorithm 6 Algorithm Detecting File Leakage by Email

1: **for** each process tree t in the OS **do**
2: **if** two processes p_1 and p_2 belong to the process tree t
and p_1 executed a **connect_to_INET** syscall issued by exim4 to an external IP address
and p_2 executed an **open** syscall issued by mailx
then
3: **return** "A file has been sent as email attachment using mailx!"
4: **end if**
5: **end for**

Figure 13: Tracing algorithm 1 pseudocode [27]

Implementation and Verification

The authors implemented the algorithms in ArchSight ESM (Enterprise Security Management which is a leading security correlation engine. Integrating the data provenance algorithms with kernel-space sensors and ESM provide a complete solution to the data leakage detection. As shown in Figure 14, a case of data leakage problem identification and its system is delineated.

Each data provenance algorithm is implemented as an ESM rule. With the help of the events captured by the sensor and the rules, The ESM correlation engine can process rules to analyze the event and detect the data leakages. The tracing algorithms are fundamental and file-centric and can help achieve full transparency in cloud computing. The above integration experiment shows the usefulness of tracing algorithms in the data leakage detection [27].

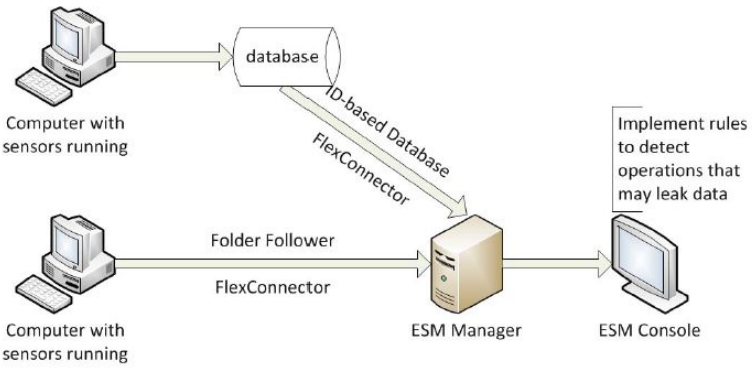

Figure 14. Overview of data provenance tracing algorithms, kernel-space sensors and ESM integration and data leakage problem [27]

3.3.6. Data Leakage Detection in Cloud Computing using BLP-allied models

In [8], authors have proposed means to identify a user who is guilty when the data from the organization is in the hands of some agent who has been compromised. This leads to data being leaked. Such a leakage is investigated in this work. A security model that targets confidentiality based issues the Bell-La Padula security model. This model enables design and analysis provisioning for securing computer systems. To this end, the model is popularly known as the data confidentiality model. This is because this model builds on developing data confidentiality by working on similar issues and giving access to classified information.

This model is based out of a state chain mechanism where the computer states follow a state transition.These states also have rules which need to be followed. These are well laid out security policies which determine the access modes and security clearance rules. These clearance schemes are expressed as a Lattice. Another model which works on data integrity is called the Biba-Integrity model. In this model the information system is divided in an object oriented approach. These algorithms are per se determined based on computational speed, complexity and tenability. The experimental factors are analyzed based on results obtained from previous studies and files used. AES has a lesser time of computation that DES. RSA takes a lot higher encryption time, also there is additional memory use (which adds up in the space complexity). All of the aforementioned techniques are useful for real time encryption. Major considerations in all of these are the length of the key, cipher, block size and analytical resistance of the crypts developed. Further security, prediction key and ASCII printable characters, time required for possible keys are all dependent on the modelling. Specifically, DES follows a key distribution based approach and has certain key management limitations, however RSA has larger overheads in terms of time and encryption operation. Therefore in this proposed model authors provide a solution for the data leakage problem based on the Bell-LaPadula model for securing infrastructure and water marking the data.

Proposed Algorithm:

For allowable access mode, a classification object is specified for a set of subjects such as:

Object O. i.e.S=(S1, S2, S3,..Sn),O=(O1, O2, O3,..On)

Both S and O are used together for security policy issuance by combining and creating a pair. The Security Model works with triplets such as the following :

Z=Triplets(S=subject, O=object, A=attribute),

The security level definition is based on the pair of(C, S).

C= classification being in the set of ‘’Public, Confidential, Secret, Top secret’’.

S= category set following Military,Air force, Defense, R and D.

Further, the precedence goes as follows (c1,s1) > (c2,s2) iff c1, c2 and S1, S2. Level-1 dominates Level-2 because the rules follow the form of a lattice.

In Bell-LaPadula model basic rules followed are as given below:

1. Reading down (NRU): ‘’A subject S can only have read access to objects O with a security level L is below the subject’s current clearance level’’. The precedence of information prevents the subjects from gaining access at random.
2. Writing up (NWD):’’A subject S has only write access to objects O with security level L is higher than its current clearance level’’. This makes a subject pass information from a higher level to a lower level.
3. Simple Security Property: (NRU- No Read Up) “ subject S at a given security level may not read an object O at a higher security level’’.
4. Append-Only: ’’The subject can only write to the object but it cannot read’’.
5. Execute-Only: ’’The subject can execute the object but can neither read nor write’’.
6. Read-Write: ‘’The subject has both read and writes per-missions to the object.’’ According to the BLP model the subject gains access to read as well as write.
7. Tranquility Principle: ‘’The tranquility principle of theBell-LaPadula model states that the classification of a subject or object does not change’’

Additionally, authors follow the model where they watermark the images in the stored documents and this also shows the organization's emblem to keep it safe. The intensity measurement values fall in the range as 0 to (2^24-1), For three components of color image as RED, GREEN and BLUE ranges from 0 to(2^8-1). Each character has their ASCII values falling in the range from 0 to(2^8 - 1). Finally, the water marking embeds a set of codewords in M. The calculation of water marker with some parameters i.e. cipher text etc in phase 1 and in phase 2 placement of cipher on the message. This water a mark document is sent to the client. Additionally, client id detection is done as the final step. This is divided in two steps, Phase 1 where the placement of the cipher and the authentication code word is mapped. Also, Phase 2 the verification of the message is done.

EVALUATION OF TECHNIQUES

A. Evaluation of Detection techniques:

1. Detection using Tracing algorithms:

   Pros: The algorithms were able to detect the operations such as copy, rename and file movement within a local machine in a cloud as well as it was able to detect file movement across different machines in the same cloud. These algorithms help in achieving transparency and accountability in the cloud computing environments.

   Cons: The algorithms have a few limitations like : unable to detect if a new file was created in which content from an existing file was copied, unable to distinguish if unrelated files were read and created in a sequence thus leading to false positives.

2. Detection using DWT:

   It is observed in [7] that this scheme is resistant towards several kinds of attacks. Several attempts that can be made in order to extract the watermark can be Gaussian noise attack, Gaussian blur attack, Salt & pepper noise attack, and Speckle noise attack. While these attacks will be unable to completely remove a watermark, they can be used to try and damage the watermark so that it is no longer recognizable. DWT watermarking scheme tackles all these attacks efficiently. One such measure to identify if the watermark has been damaged is to compare the correlation coefficient values of the extracted watermark before and after the attack.

Attacks	Watermarked Image Before Attack	Watermarked Image After Attack
Gaussian blur	0.9869	0.9893
Gaussian noise	0.7257	0.7371
Salt&pepper noise	0.8961	0.9191
Speckle noise	0.9151	0.9347

Table VI. Performance against different attacks[7]

As are clearly visible, the correlation values of the pixels are hardly affected by the attempted attacks to destroy the watermark. Therefore, if all copies of images stored in the cloud are watermarked using the DWT watermarking scheme, any leaks can be detected by extracting the watermarks of the leaked data and comparing it with the image used to watermark it in the first place. This scheme works well for keeping the ownership of image data secure. If any intruder tries to claim ownership of any leaked data, this scheme can be used to verify the true owner of the leaked data. Also, if the intruder tries to sell this data or use it publicly, the rightful owner will be able to track and claim rightful ownership by extracting the watermark. On the other hand, if the intruder only plans on using the information extracted from the leaked data just to gain knowledge, this scheme is unable to provide a method in order to tackle that issue.

3. Dynamic Data Leakage Detection model based approach: The main motive for this algorithm was to make the data more secure in the mapreduce framework and to get notified when a data leakage occurs. The results of the algorithm show the processing time of the data has improved on the basis of distribution of equal quantity of data and also approximately equal response time from each Virtual Machine. The overall performance of the system is improved by reducing the data by approximately 70-80% by using map reduce function. The algorithm has also improved the probability of finding a guilty agent by using the s-max algorithm.

4. Detection and prevention using MyDLP. MyDLP is an open source data loss prevention software. It runs with multi-site configurations on network servers and endpoint computers. It allows us to monitor, inspect and prevent all outgoing confidential data from the organization in use. This is done through accurate pattern matching detectors called content blades. It identifies sensitive data using deep content analysis. It is used to prevent data leakage as well, it blocks all outgoing confidential data. External systems like printers are also blocked from accessing confidential data.

B. **Evaluation of Defense techniques**:

1. Defense using MetaData Based Data Storage Mechanism:

Pros: This method proposes a new methodology that aims at making the data invaluable to the hacker rather than concentrating on restricting the hacker. It makes data valuable only during acquisition and while the data is being updated. The design of the model segregates the data and further breaks it down into fragments while keeping the key elements seperated.

Cons: The main drawback is that the DME in the model has to be initially configured which takes a lot of effort and then migration of the existing data to the new model. The changes to the existing conventional database engines are unavoidable because there will be an inherent need for plugging in the DME and the database runtime migration environment to these engines. Fragment of data also incurs a huge cost. This cost includes the cost of fragmentation of data while storage and also the cost of forming the data at runtime from the fragmented data. In addition to this fragmentation, a proper encryption technique has to be used to provide additional security. This encryption can be done only to data that is fragmented as 'sensitive' by the DME. This reduces the cost of encryption of the entire database.

2. Defense using Multi-tier Security Approach:

   Proposed algorithm in [21] has a lot of pros for it to be not considered. It has so many benefits compared to the existing and prevalent technique of Multilevel Authentication algorithm. Our primary goal here is to improve the security and with this 3-level authentication we will surely be able to do that, since we have Secure Keyboard, OTP, Pin code and Image based verification system to robustize the security process. We also see some results that compare the overall performance of the EXISTING v/s PROPOSED algorithm. Apart from security we compare the results of Data Center Processing Time, Total Virtual Machine Cost, Total Data Transfer Cost and Overall Response Time. In most of these parameters Proposed algorithm outperforms the existing technique.

3. Defence using Fog Computing:

   Pros: This method uses a novel approach by the combination user behavioural profiling and launching disinformation attacks using decoy files on a suspected intruder. This approach has high probability to detect a malicious intruder and protect the real users data from insider attacks. Also, this approach can inform the real user about a suspicious activity, so that the user can do the necessary action (i.e., changing password of the account, moving sensitive information to protected folder).

   Cons: In this method, a real user is also prone to disinformation attacks since a real user can unintentionally open a decoy file which can trigger the disinformation attack. Also, over the course of time, the machine learning model's accuracy of classifying intruders access patterns might decline because the real user's access pattern becomes increasingly complex.

4. Comparison of Encryption algorithms.

Encryption is an important technique for protection of data from unauthorized access. The most commonly used algorithms used for encryption and decryption are RSA, AES, DES and Blowfish. The execution time required for the RSA algorithm is highest. Also, the space required for this algorithm is large compared to other modern algorithms. A public key is used for RSA, which causes lower security of data and thus providing security only to users and not the service provider. Due to these disadvantages, RSA is suitable only when dealing with small data. Another algorithm used is the Blowfish algorithm. It takes less execution time compared to RSA. It requires the least space compared to other algorithms and requires less than 5kb for execution. It uses the same key of variable length for encryption and decryption and the key is not public, making it both flexible and secure and also capable of providing security for both client and service provider. DES is a block cipher and decrypts data in blocks of size 64bits. DES is faster compared to blowfish but takes a large amount of space for execution. The size of the key used is 56 bits and is the same for encryption and decryption and can provide security to both client and provider. Advanced encryption standard is currently the most widely used algorithm. It is the fastest encryption algorithm and requires very less memory for execution. The size of the key used can be up to 256 bits providing highest security and best authentication for the data.

CONCLUSIONS AND RECOMMENDATIONS

Currently cloud services pose a significant danger to data leakage or data breach[1]. An enhanced attack occurs with each evolved protection mechanism against these types of assaults. Mechanisms of protection for stopping data breaches are not always successful alone. It is highly recommended that various mechanisms be combined to create hybrid protection mechanisms, with different layers of cloud computing. This is because hybrid solutions typically involve criticality measures as well as identification measures. As opposed to the state of the art techniques that only consider one of the two such as some assurance techniques focus only on the static establishment and not on the enforcement of the defense mechanism, making this dynamic with a hybrid model is the way to go forwarded. Hence, adaptation to newer techniques and efficient trade offs are the two important metrics which have allowed us to come to this conclusion. Historical examples of different forms of data leakages are provided in this paper[1]. We also investigated the effect on the cloud environment of various forms of protective and detective mechanisms. Ultimately, in cloud-based applications, we analyzed and described suggested security or detective mechanisms. The paper[1] provides a description of challenges in cloud computing problems .

Cloud computing is becoming one of the key words of the IT industry. As the usage of cloud infrastructure is increasing, the security of cloud systems has become a vital element of cloud computing systems. The various security issues related to cloud infrastructure are confidentiality, integrity, availability and privacy issues. There are two categories of data leakage discussed in the report[2]. Direct and Indirect data leakage is discussed. Detection of this leakage of data becomes difficult as security is introduced in the system. The challenges that are faced for detecting leakage of data are caused due to Encryption. Encryption of data ensures confidentiality, authenticity, and integrity of the data[15]. It also makes it difficult to identify the data leaks occurring over encrypted channels. Another challenge faced in detection of data leakage is Access Control. Access control is suitable for data at rest, it is difficult to implement for data in transit and in use. Semantic Gap in DLP is another challenge faced in detection of data leakage. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching, or access control scheme cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios. There are existing systems to detect leakage of data. Watermarking is an existing technique used to detect leakage of data. We have proposed two algorithms that can be used to detect data leakage. Evaluation of Explicit Data Request Algorithms is

a technique that can improve the detection of the guilty agent and to evaluate our e-optimal algorithm relative to random allocation. Another method that can be used is Evaluation of Sample Data Request Algorithms. In this method, object sharing is not explicitly defined by their requests. The distributor is "forced" to allocate certain objects to multiple agents only if the number of requested objects exceeds the number of objects in set T. There are various types of modules that facilitate data detection. Data Allocation Module is Important to know how the distributor can "intelligently" give data to agents in order to improve the chances of detecting a guilty agent. In Fake Object Module, Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data in order to improve their effectiveness in detecting guilty agents. The Optimization Module is the distributor's data allocation to agents that has one constraint and one objective.In Data Distributor Module, the gives sensitive data to a set of supposedly trusted agents. The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Agent Guilt Module helps in determining fake agents using fake objects and probability.

We have also studied about the causes of data leakage in cloud infrastructure. They are categorized into Internal and External Threats. Internal Threats are caused due to Intentional and unintentional data leakage. Intentional Data Leakage is due to outsourcing unauthorized data by the internal users. Unintentional Data Leakage occurs when an authorized user sends confidential information to an unauthorized user by mistake. There are six external threats. Data theft by intruders includes the intruders stealing sensitive information. SQL injection attacks mainly aim at stealing data from the database of the websites using SQL server in the backend. Malware attacks are caused when a system gets infected with malware. This results in the loss of private data. Destroying any sensitive information before disposing the data is an important step and it is called Dumpster diving. Phishing is a technique used by hackers to gain information of the users by sending mails. Physical burglary is caused due to weak physical security that can be exploited which may lead to loss of devices which contains sensitive information.

We have also studied about threats and measures that are used to counter the threats in this project. From our study and research we have found six different important categories of security measures in cloud infrastructure[18]:

1. **Embedded Security** contains the ability of a system to connect to an external local network using high quality tools. The isolation provided by the virtual machines prevent stray user interference thereby providing strong security. The deployment of these virtual machines can

cause failure in security. There should be measures taken by the companies providing these infrastructure and when uploading them in public. The Virtual machine should be controlled to update the resource requirement and changes in host parameters should be monitored carefully.

2. **Software applications** are a bottleneck of security. Software applications act as points of vulnerability and determine how secured is the system. Both front end and back end security should be taken into consideration. The software code should be monitored and maintained regularly as vulnerabilities arise due to bad maintenance of code.
3. **Client management** is an integral property of cloud computing security. Protection of client information is called client management. Some cloud service providers fail to provide such solutions that leads to poor customer experience, which leads to poor security. Authentication and accounting is a major determining factor for selection of cloud service providers.
4. **Data stored in the cloud** plays a vital role in the security of cloud computing systems. All applications and data should be put under the firewall for preventing security attacks. Deployments of data warehouses brings up the need for high security and shows the quality of a service provider.
5. **Cluster Computing** is the collection of multiple Virtual Machines or servers looped together to maintain application execution in a parallel or serial fashion. Virtual Machines make it important to consider what is the
6. **Operating system** used. Security issues that are caused due to such heterogeneity need to be further analyzed and updated.

There are other losses caused due to natural disasters and threats to cloud computing services. The metrics such as reliability, usability, and extensibility are used to measure the quality of cloud service providers. Incomplete authorization and lack of accounting can lead to issues in reliability. Encrypting the data can lessen the abnormalities and effects of a data breach and can increase the quality of service provided. Third party intrusions are possible if the institutions are giving away their credentials to third parties for application developments. Malicious insiders can propagate falsified information that can lead to data insecurity or loss.Cloud hijacking is another issue in which a malicious insider can steal the credentials to edit vital information.

We have studied various data mitigation techniques[14]. Source content management is a technique used to avoid data leakage using secured channels. Reputation system assigns a score for each email sender reducing the number of unwanted emails and indirectly reducing phishing and spamming. TThe users

should be provided with only the necessary applications. Restricting the usage of USB drives and other external storage devices can also help to solve threats of internal attacks and hijacking. The data that is sent to various insiders should be protected by the level of access allowed to the particular person. Finally, we can block malicious websites that are obtained through browsing history to reduce phishing attacks on the system.

We also studied about different kinds of Defense Mechanisms involved in securing the cloud. The first method is **Metadata based Data Storage Model [24]**. This method proposes a new methodology that aims at making the data invaluable to the hacker rather than concentrating on restricting the hacker. It makes data valuable only during acquisition and while the data is being updated. The design of the model segregates the data and further breaks it down into fragments while keeping the key elements seperated.

This method is very well explained with the use of credit card scenarios. This method is effective when a database is being created from scratch and Data Migration Environment (DME) has been used for moving existing data to the cloud. However the DME model takes a huge amount of effort as it first needs to be configured and then the migration needs to take place. Additionally, the fragmentation of data also incurs a huge cost as the data first needs to be fragmented at storage and then formed during the runtime.

The second method is **Mitigating Insider Data Theft Attacks in Cloud:** This method uses a different technique for securing data called Fog Computing. This method uses the strategy of launching numerous disinformation attacks against the malicious intruders such that they are unable to distinguish between real data and fake data. Here the combination of two techniques are used i.e. User Behaviour Profiling and Decoy Technology. This combination improves the security of confidential data and it becomes possible to detect the data accesses that are not authorized on the local file system by masqueraders. A Support Vector Machine (Machine Learning Model) is used to detect anomalies in user behaviours. The traps are placed in the highly-conspicuous locations in the file system. This methodology provides several advantages such as Detecting the masquerading activity, Confusing the intruder and impacting the intruder with bogus information. Also the results of this method show that it is accurate enough to detect the unauthorized access and can be used to monitor and protect real user's data in the cloud computing environment.

The third method proposed is a new approach called **Multi-tier Security approach[21]**. It not only discusses the algorithm, but it's performance as well as how it robustizes the security. Firstly, in [21] we

see there are several procedures in place to avoid these attacks. One of the most commonly used measures is OTP. This paper refines the encryption method of OTP along with usage of top secret PIN to provide proper authentication and security. As we saw how effective Festal Network Process can be if used to secure OTP. In the report we consistently see different authors describing the importance of watermarking algorithms and online verification via generating E-Id. [21] further lists down the parameters that can be used to identify whether the user is genuine or not. The paper further breaks down the Security Components into - Secure Keyboard, Login Process, OTP and Image Based Verification System.The above components can be used at different levels of authentication and collectively provide robust security. Furthermore, the author goes on to describe entities involved in architecture as follows: User, Master Server - 1, Master Server - 2 , Master Server - 3, Authentication & Cloud Server. In this paper [21], it takes above 6 entities and the 4 components of security described above and describes the proposed algorithm. As we proceed further, it compares various results from the proposed algorithm and results from multilevel authentication algorithms which is currently very prevalent in the current scenario of cloud computing security mechanisms. In the results, initially we compare data center processing time, virtual machine and data transfer cost for both existing and proposed algorithms. Later we look at response times for both algorithms as well. Looking at these results we can clearly conclude that proposed algorithms of Multi-tier security approach easily outperforms existing prevalent Multi-factor authentication algorithms.

Further in the report we discuss Detection techniques. In the report we have described extensively the rigorous research we did on the detection techniques, because this is one of the most important parts of Security and Data Leakage in Cloud Computing Systems.

1) Watermarking technique is one of the most important kinds of technique. Herein, we embed a code or encryption on the distributary data information. It comes to the rescue even if any images, videos or documents get leaked, since watermarking can act as a proof of ownership.
2) Data allocation can be used to detect people who are leaking data. The distributer sends the data using data allocation strategies and adding fake objects in the data. On the basis of the assumption that the agent would again leak the data the distributor checks what fake information is out on the internet. With enough evidence legal action can be taken against agents.
3) Detection using Discrete Wavelet Transform. It splits an image into 4 sub-bands by passing an image through a pair of filters. The proposed algorithm performs 3 levels of DWT on the image which is to be watermarked (chosen by the owner of the image). The watermark can only be

removed by accessing the original image. The technique works well for keeping image data secure, but if someone has to extract knowledge just from leaked data then this technique won't be able to tackle that issue.

4) Detection and prevention using MyDLP. It identifies sensitive data using deep content analysis. It is used to prevent data leakage as well, it blocks all outgoing confidential data. External systems like printer are also blocked from accessing confidential data
5) Detection using load balancing and Map Reduce framework - As we saw in the report the load balancer, balances the load and later map reduce framework reduces the load at each and every appliance. This results in reduced data and then the distributors make this reduced data secure. In case if any data leakage takes place Distributors can help in finding the guilt agent. In the report we also show results of how the response time improves, data is reduced and is more secure than ever.

The next section deals with the overall analysis of challenges in data leakage prevention mechanisms. This covers a broad area of topics that are crucial to secure data in the cloud. Sensitive information can be spilled through the **leakage channels** such as USB, CD drives etc. It is important to have intelligent access rights to reduce the threat of data leakage. **Human activities** are always hard to estimate as they are very subjective. However different degrees of access rights can mitigate the risks. **Access Rights** are very important to recognize the clients with respect to their privileges. Another important aspect is the **Encryption and Steganography.** DLPS systems won't be able to identify a scrambled document as a touchy record by checking the encoded record with any unique information. **Data Modification** takes place by including or subtracting a number of lines before sending it across. Some Data Leakage Prevention Framework also use hashing techniques to check information leakages. **Data Classification** is used by many Data Leakage Prevention Mechanisms. As the systems would only be able to distinguish between a confidential and normal data only if it is classified. Additionally we study the Data Leakage Detection using Tracing Algorithm[26]. Here a tracing algorithm is used to secure data in the cloud. In order to facilitate data tracing in the cloud, A logging mechanism and a powerful Security Information and event management has been used. The tracing algorithm has been implemented using 5 layers i.e. Systems, Data, Workflows, Laws and regulations and Policy. Different types of Rule based Provenance Tracing Algorithms are used for local machines and Cross-machine. The **Local Tracing algorithm** aims at detecting file operations that could cause data leakage. These mainly detect File Copying, File Renaming and File movement.

Cross Machine Tracing algorithms [27] take into account that there are multiple processes and to deal with this processes are organized in the form of tree structure and any actions on the processes trees are detected. These algorithms are implemented in ArchSight ESM (Enterprise Security Management) as an ESM rule to detect data leakages. The overall results show that these algorithms are very useful in detecting data leakages.We have also studied data leakage detection using BLP- allied models[8]. These models help us to find the guilty agents in case there is a data leakage in the system. The Bell-La Padula enables design and analysis that are carried out to provide security for the data. This model is built on the basis of development of data confidentiality and providing required access to classified information. We have also studied about the rules which should be followed. We have also studied the Biba-Integrity model, which works on integrity of data. The model is based on division which is object oriented and objects and subjects work together to provide states for interaction the transition from one state to another following Markov chain models thus satisfying the security of the objects.AES and RSA algorithms are used for encryption of data in real time. AES takes less computation time . RSA requires a lot of execution time and takes up extra storage space. We have also studied a proposed algorithm based on the Bell-LaPadula model for watermarking and security. The model works with security triplets which consist of subject, object and attributes. The images in the data are watermarked and also have the logo of the company to ensure its security. The intensity is measured, the characters are color coded based on the ASCII value. The watermarking finally adds codewords in data, whose calculation is carried out using some parameters. The client id detection is the final step which consists of placement of cipher and authentication code and verification of message. Thus, in this way we have studied and expounded the major detection and defence approaches in cloud computing.

REFERENCES

1. Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2016). A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, *62*, 137-152.
2. Bollam, N., & Malsoru, M. V. (2011). Review on Data Leakage Detection. *International Journal of Engineering Research and Applications (IJERA)*, *1*(3), 1088-1091.
3. Barona, R., & Anita, E. M. (2017, April). A survey on data breach challenges in cloud computing security: Issues and threats. In *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (pp. 1-8). IEEE.
4. Shabtai, A., Elovici, Y., & Rokach, L. (2012). *A survey of data leakage detection and prevention solutions*. Springer Science & Business Media.
5. Purohit, B., & Singh, P. P. (2013). Data leakage analysis on cloud computing. *International Journal of Engineering Research and Applications*, *3*(3), 1311-1316.
6. Shobana, V., & Shanmugasundaram, M. (2013). Data leakage detection using cloud computing. *International Journal of Emerging Technology and Advanced Engineering*, *3*(1), 111-115.
7. Ingale SP, Dhote CA (2016) Digital watermarking algorithm using DWT technique. International Journal of Computer Science and Mobile Computing, IJCSMC 5(5):1–9
8. Kumar, N., Katta, V., Mishra, H., & Garg, H. (2014, November). Detection of data leakage in cloud computing environment. In *2014 International Conference on Computational Intelligence and Communication Networks* (pp. 803-807). IEEE.
9. Chhabra, S., & Singh, A. K. (2016, December). Dynamic data leakage detection model based approach for MapReduce computational security in cloud. In *2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)* (pp. 13-19). IEEE.
10. Pemmaraju, S., Sushma, V., & Sagar, K. D. (2014). Data Leakage Detection using Cloud Computing. *Global Journal of Computer Science and Technology*.
11. Kowsik, R., & Vignesh, L. (2016, March). Mitigating insider data theft attacks in the cloud. In *2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM)* (pp. 561-567). IEEE.
12. Bollam, N., & Malsoru, M. V. (2011). Review on Data Leakage Detection. *International Journal of Engineering Research and Applications (IJERA)*, *1*(3), 1088-1091.
13. Carlson, F. R. (2014). Security analysis of cloud computing. *arXiv preprint arXiv:1404.6849*.

14. Brindha, T., & Shaji, R. S. (2015, December). An analysis of data leakage and prevention techniques in cloud environment. In *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)* (pp. 350-355). IEEE.
15. Barona, R., & Anita, E. M. (2017, April). A survey on data breach challenges in cloud computing security: Issues and threats. In *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (pp. 1-8). IEEE.
16. Kale, S. A., & Kulkarni, S. V. (2012). Data leakage detection. *International Journal of Advanced Research in Computer and Communication Engineering*, *1*(9), 668-678.
17. Amara, N., Zhiqui, H., & Ali, A. (2017, October). Cloud computing security threats and attacks with their mitigation techniques. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 244-251). IEEE.
18. Alhenaki, L., Alwatban, A., Alamri, B., & Alarifi, N. (2019, May). A Survey on the Security of Cloud Computing. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-7). IEEE.
19. Indira, B., & CH, D. M. A LITERATURE REVIEW ON DATA LEAKAGE DETECTION IN CLOUD COMPUTING.
20. Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. *International journal of engineering research and applications*, *3*(4), 1922-1926.
21. Kirar, A., Yadav, A. K., & Maheswari, S. (2016, November). An efficient architecture and algorithm to prevent data leakage in Cloud Computing using a multi-tier security approach. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)* (pp. 271-279). IEEE.
22. Sabahi, F. (2011, May). Cloud computing security threats and responses. In *2011 IEEE 3rd International Conference on Communication Software and Networks* (pp. 245-249). IEEE.
23. Anitha, R., & Mukherjee, S. (2014). Data security in cloud for health care applications. In *Advances in computer science and its applications* (pp. 1201-1209). Springer, Berlin, Heidelberg.
24. Subashini, S., & Kavitha, V. (2011, October). A metadata based storage model for securing data in cloud environments. In *2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 429-434). IEEE.
25. Hussein, N. H., & Khalid, A. (2016). A survey of Cloud Computing Security challenges and solutions. *International Journal of Computer Science and Information Security*, *14*(1), 52.

26. M. Ben-Salem and S. J. Stolfo, “Modeling user search behavior for masquerade detection,” in Proceedings of the 14th International Sympo-sium on Recent Advances in Intrusion Detection. Heidelberg: Springer, September 2011, pp. 1–20.
27. O. Q. Zhang, R. K. L. Ko, M. Kirchberg, C. H. Suen, P. Jagadpramana and B. S. Lee, "How to Track Your Data: Rule-Based Data Provenance Tracing Algorithms," 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, 2012, pp. 1429-1437.